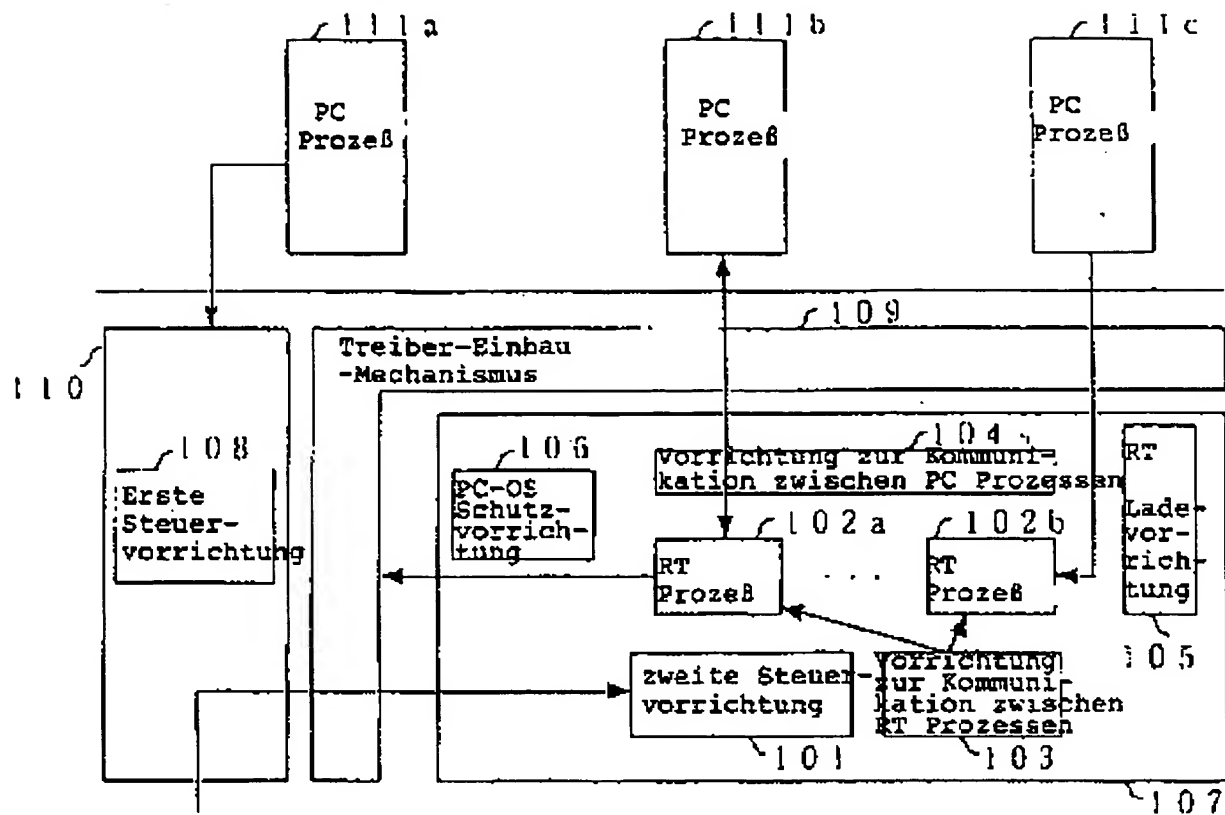


AN: PAT 1998-078206
TI: Real-time control system for general-purpose personal computer has first and second controllers for association of CPU capacity with user and real-time processes respectively and for controlling corresp. processes
PN: **DE19728989-A1**
PD: 15.01.1998
AB: The system has a first controller for the association of CPU capacity with user processes and for controlling the processes. A drive installation mechanism installs a device driver to perform input-output processes for equipment. A real-time control mechanism accesses processes requiring a real-time response to an interrupt from a device such as a device driver from an operating system. Real-time processes are formed in association with the devices and a second controller associates CPU capacity with the real-time processes and controls the processes.; USE - For PCs. ADVANTAGE - Hardware costs are reduced and system can be flexibly adapted to improved PC operating system. Enables real-time and PC processes to co-exist.
PA: (MITQ) MITSUBISHI DENKI KK;
(MITQ) MITSUBISHI ELECTRIC CORP;
IN: KUROSAWA H;
FA: **DE19728989-A1** 15.01.1998; CN1172986-A 11.02.1998;
JP10021094-A 23.01.1998; KR98010769-A 30.04.1998;
TW349205-A 01.01.1999; KR265679-B1 15.09.2000;
CO: CN; DE; JP; KR; TW;
IC: G05B-015/00; G06F-009/46; G06F-013/10;
MC: T01-F02C1; T01-J08A1; T01-K; T06-A07A;
DC: T01; T06;
FN: 1998078206.gif
PR: JP0177898 08.07.1996; KR0031230 05.07.1997;
FP: 15.01.1998
UP: 23.11.2001

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)



BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

(19) BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

(12) Offenlegungsschrift

(10) DE 197 28 989 A 1

02 P 03971

(51) Int. Cl.⁶:
G 05 B 15/00
G 06 F 9/46
G 06 F 13/10

(21) Aktenzeichen: 197 28 989.4
(22) Anmeldetag: 7. 7. 97
(43) Offenlegungstag: 15. 1. 98

DE 197 28 989 A 1

(30) Unionspriorität:

8-177898 08.07.96 JP

(71) Anmelder:

Mitsubishi Denki K.K., Tokio/Tokyo, JP

(74) Vertreter:

PFENNING MEINIG & PARTNER, 80336 München

(72) Erfinder:

Kurosawa, Hisayoshi, Tokio/Tokyo, JP

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Echtzeitsteuersystem

(57) System, umfassend einen Mechanismus für ein PC-OS, um einen Eingabe/Ausgabe-Gerätetreiber aufzurufen, eine zweite Steuervorrichtung zur Verteilung von erworbenen CPU-Nutzrechten der Echtzeitprozesse, eine Ladevorrichtung als ein Echtzeitprozeß und eine Vorrichtung zur Kommunikation zwischen Prozessen.

DE 197 28 989 A 1

HINTERGRUND DER ERFINDUNG

1. Bereich der Erfindung

Diese Erfindung bezieht sich auf einen Echtzeitsteuermechanismus für Personalcomputer, der für die Koexistenz eines Prozesses, der Echtzeiteigenschaften erfordert, und eines Prozesses, der auf einem Allzweckpersonalcomputer betrieben wird, gedacht ist.

2. Beschreibung des Standes der Technik

Mit dem rasch weitverbreiteten Gebrauch von Personalcomputern (PCs) wurde in den letzten Jahren auch gefordert, die verschiedenen Softwaremöglichkeiten, die auf PCs vorhanden sind, in einem Echtzeitsystem gut zu nutzen, das früher als geschlossen betrachtet wurde.

Um die Softwareressourcen auf PCs gut zu nutzen, müssen ein RT-Prozeß, der auf einem Echtzeitsystem läuft, und ein PC-Prozeß, der auf einem PC läuft, in Zusammenarbeit miteinander laufen. Bisher wurden die folgenden zwei Verfahren verwendet:

Herkömmliches Beispiel 1

Beispielsweise ist ein Verfahren, das auf Seite 201 von "Keisoku to Seigyo Band 34, Nr. 3" (erschienen im März 1995) beschrieben ist, als erstes Verfahren verfügbar.

Bei diesem Verfahren sind eine CPU 1 (1301) zum Betrieb eines PC-OS (110) und eine CPU 2 (1302) zum Betrieb eines Echtzeit-OS (1304) getrennt installiert zum Betrieb von RT-Prozessen (102) unter der Kontrolle des Echtzeit-OS (1304) und PC-Prozessen (111) unter der Kontrolle des PC-OS (110), wie in Fig. 15 gezeigt.

Die RT und PC Prozesse tauschen Daten miteinander über einen gemeinsamen Speicher (1303) aus, auf den von der CPU 1 und der CPU 2 zugegriffen werden kann.

Herkömmliches Beispiel 2

Beispielsweise ist ein Verfahren als das zweite Verfahren verfügbar, das auf Seite 142 von "Interface für Juni 1996" (CQ Shuppan) beschrieben ist.

Bei diesem Verfahren wird ein Echtzeit-OS (1304) als Kern benutzt und ein PC-OS (1401) wird darauf in der Form eines Emulators, wie in Fig. 16 gezeigt, installiert.

RT-Prozesse (102) benützen unmittelbar Dienste, die durch das Echtzeit-OS (1304) zur Verfügung gestellt werden, und PC-Prozesse (111) laufen auf dem PC-OS-Emulator (1401).

Da der herkömmliche PC-Echtzeitkontrollmechanismus so aufgebaut ist, müssen extra Hardware wie CPUs und gemeinsamer Speicher zum Betrieb des Echtzeit-OS und von RT-Prozessen installiert werden, wobei die Kosten erhöht werden.

Wenn der PC-OS-Emulator für die Konfiguration angenommen wird, kann ein Anstieg der Kosten durch Installation von zusätzlicher Hardware unterdrückt werden, jedoch muß der gesamte PC-OS-Emulator jedesmal geändert werden, wenn das PC-OS auf eine höhere Version verbessert wird, aufgrund der Notwendigkeit, der neuesten PC-OS-Version zu folgen, was in einem Anstieg in Aufwendungen für Softwareentwicklung resultiert.

Es ist daher ein Ziel der Erfindung, einen PC-Echtzeitsteuermechanismus zur Verfügung zu stellen, der Hardwarekosten unterdrücken kann und flexibel mit der Verbesserung eines PC-OS auf eine höhere Version umgehen kann, um ein System zur Verfügung zu stellen, in welchem RT und PC Prozesse koexistieren.

Hierzu wird gemäß eines ersten Aspektes der Erfindung in einem Betriebssystem, das eine erste Steuervorrichtung zur Zuweisung von CPU-Kapazitäten zu Anwendungsprozessen und zur Steuerung der Anwendungsprozesse als auch einen Treiber-Einbau-Mechanismus zum Einbau eines Gerätetreibers zur Durchführung von Eingabe/Ausgabe-Bearbeitung für Geräte umfaßt, ein Echtzeitkontrollsystem zur Verfügung gestellt, das Verarbeitungen bildet, die Echtzeit-Reaktionsfähigkeit auf einen Interrupt von einem Gerät erfordern, wie ein Echtzeitsteuermechanismus, der wie ein Gerätetreiber von dem Betriebssystem angesprochen wird, dadurch charakterisiert, daß der Echtzeitsteuermechanismus Echtzeitprozesse umfaßt, die in einer Bearbeitungskorrespondenz mit den Geräten gebildet werden, und eine zweite Steuervorrichtungen zur Zuordnung der CPU-Kapazitäten zu den Echtzeitprozessen und zur Steuerung der Echtzeitprozesse umfaßt.

Bei dem Echtzeitsteuersystem nach dem ersten Aspekt der Erfindung umfaßt in einem zweiten Aspekt der Erfindung das Echtzeitsteuersystem eine Vorrichtung zur Kommunikation zwischen Echtzeitprozessen zur Übertragung von Daten zwischen den Echtzeitprozessen.

Bei dem Echtzeitsteuersystem nach dem ersten Aspekt der Erfindung schließt in einem dritten Aspekt der Erfindung der Echtzeitsteuermechanismus weiterhin eine Vorrichtung zur Kommunikation zwischen Prozessen zur Übertragung von Daten zwischen dem Anwendungsprozeß, der auf dem Betriebssystem läuft, und dem Echtzeitprozeß ein.

Bei dem Echtzeitsteuersystem nach dem ersten Aspekt der Erfindung schließt in einem vierten Aspekt der Erfindung der Echtzeitsteuermechanismus weiterhin einen Lademechanismus für Echtzeitprozesse zur Eintragung des Anmeldeprozesses, der auf dem Betriebssystem läuft, als Echtzeitprozeß ein.

Bei dem Echtzeitsteuersystem nach dem ersten Aspekt der Erfindung umfaßt in einem fünften Aspekt der Erfindung der Echtzeitsteuermechanismus weiterhin eine Schutzvorrichtung für das Betriebssystem ein, um eine Dienstfunktion zu ermöglichen, die durch das Betriebssystem für den Gerätetreiber zur Verfügung gestellt wird, um aus dem Echtzeitprozeß benutzt zu werden.

Bei dem Echtzeitsteuersystem nach dem ersten Aspekt der Erfindung umfaßt in einem sechsten Aspekt der Erfindung der Echtzeitsteuermechanismus weiterhin eine Vorrichtung zur Überwachung von Nutzungsrechten, um die Zeit zu messen und festzuhalten, die die zweite Steuervorrichtung die CPU-Kapazitäten von dem Betriebssystem besetzt und um die CPU-Kapazitäten abhängig von dem Ergebnis der Messung zwangsweise an das Betriebssystem zurückzugeben.

In einem siebten Aspekt der Erfindung umfaßt das Echtzeitsteuersystem nach dem ersten bis sechsten Aspekt der Erfindung weiterhin eine Vorrichtung zur Konkurrenzkontrolle, wenn es eine Mehrzahl von Echtzeitsteuermechanismen gibt, um das Master-Slave-Verhältnis oder das Prioritätsverhältnis zwischen den Echt-

zeitsteuermechanismen einzustellen.

Die obigen und andere Ziele und Merkmale der vorliegenden Erfindung werden aus der folgenden Beschreibung in Verbindung mit den begleitenden Zeichnungen deutlicher werden.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

In den begleitenden Zeichnungen ist

Fig. 1 ein Systemblockdiagramm um eine erste Ausführungsform der Erfindung zu zeigen;

Fig. 2 ein Flußdiagramm, um den Interruptdienst eines PC-OS in der ersten Ausführungsform der Erfindung zu zeigen;

Fig. 3 ein Flußdiagramm, um einen Gerätetreiber-Eingangs/Ausgangsanforderungsprozeß des PC-OS in der ersten Ausführungsform der Erfindung zu zeigen;

Fig. 4 ein Flußdiagramm, um die Verarbeitung einer zweiten Steuervorrichtung zu zeigen, wenn ein RT-Prozeß als eine Routine des Echtzeitsteuermechanismus in der ersten Ausführungsform der Erfindung installiert wird;

Fig. 5 ein Flußdiagramm, um die Verarbeitung der zweiten Steuervorrichtung zu zeigen, wenn ein RT-Prozeß eine besondere Umgebung in der ersten Ausführungsform der Erfindung besitzt;

Fig. 6 ein Flußdiagramm, um die Verarbeitung der Vorrichtung zur Kommunikation zwischen RT-Prozessen in der ersten Ausführungsform der Erfindung zu zeigen;

Fig. 7 ein Flußdiagramm, um die Verarbeitung der Vorrichtung zur Kommunikation zwischen RC-Prozessen in der ersten Ausführungsform der Erfindung zu zeigen, für einen PC-Prozeß, der Daten empfängt;

Fig. 8 ein Flußdiagramm, um die Verarbeitung der Vorrichtung zur Kommunikation zwischen PC-Prozessen in der ersten Ausführungsform der Erfindung zu zeigen für einen PC-Prozeß, der Daten überträgt;

Fig. 9 ein Systemblockdiagramm, um eine zweite Ausführungsform der Erfindung zu zeigen;

Fig. 10 ein Flußdiagramm, um die Verarbeitung der Vorrichtung zur Überwachung von Nutzungsrechten in einer zweiten Ausführungsform der Erfindung zu zeigen;

Fig. 11 ein Flußdiagramm, um die Verarbeitung der zweiten Steuervorrichtung in der zweiten Ausführungsform der Erfindung zu zeigen;

Fig. 12 ein Flußdiagramm, um ein anderes Verarbeitungsbeispiel der Vorrichtungen zur Überwachung von Nutzungsrechten in einer zweiten Ausführungsform der Erfindung zu zeigen;

Fig. 13 ein Systemblockdiagramm, um eine dritte Ausführungsform der Erfindung zu zeigen;

Fig. 14 ein Flußdiagramm, um die Verarbeitung der Vorrichtung zur Konkurrenzkontrolle in der dritten Ausführungsform der Erfindung zu zeigen;

Fig. 15 ein Systemblockdiagramm, um ein herkömmliches Echtzeitsteuersystem zu zeigen; und

Fig. 16 ein Systemblockdiagramm, um ein weiteres herkömmliches Echtzeitsteuersystem zu zeigen.

GENAUE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

Im folgenden wird eine Beschreibung der Ausführungsformen der Erfindung unter Bezug auf die begleitenden Zeichnungen in größerer Genauigkeit gegeben werden.

Ausführungsform 1

Eine erste Ausführungsform der Erfindung wird unter Bezug auf die Fig. 1 bis 8 erörtert.

Fig. 1 ist ein Blockdiagramm, um eine Systemkonfiguration in der ersten Ausführungsform zu zeigen.

In Fig. 1 bezeichnet die Zahl 101 eine zweite Steuervorrichtung zur Verteilung der CPU-Nutzungsrechte auf die RT-Prozesse 102a und 102b und zur Verwaltung des RT-Prozeßausführungszustandes als eine zuvor registrierte Interrupt-Dienstroutine, die durch ein PC-OS als Antwort auf einen Interrupt von irgendeinem Gerät anders als eine CPU aufgerufen wurde.

Die Ziffer 103 bezeichnet eine Vorrichtung zur Kommunikation zwischen RT-Prozessen zur Ermöglichung von Datenaustausch und Synchronisationsverarbeitung zwischen den RT-Prozessen 102a und 102b und die Ziffer 104 bezeichnet eine Vorrichtung zur Kommunikation zwischen PC-Prozessen zur Ermöglichung von Datenaustausch und Synchronisationsverarbeitung zwischen den RT- und PC-Prozessen.

Die Ziffer 105 bezeichnet eine RT-Ladevorrichtung zur Ausführung eines Prozesses als ein RT-Prozeß, der vorbereitet und überprüft wurde als ein PC-Prozeß. Die Zahl 106 bezeichnet eine PC-OS-Schutzvorrichtung um es einer PC-OS-Funktion, die im wesentlichen für einen Gerätetreiber durch das PC-OS zur Verfügung gestellt wird zu ermöglichen als RT-Prozeß verwendet zu werden.

Die Zahl 107 bezeichnet einen PC-Echtzeitsteuermechanismus, der als ein Gerätetreiber von dem PC-OS betrachtet wird und der die Vorrichtungen und RT-Prozesse 101-106 enthält. Die Ziffer 108 bezeichnet eine erste Steuervorrichtung, die in den PC-OS enthalten ist, zur Kontrolle der Ausführungsordnung der PC-Prozesse, die Ziffer 109 bezeichnet einen Treiber-Einbau-Mechanismus, um einen Rahmen für den Bau eines Gerätetreibers in das PC-OS zur Verfügung zu stellen, und die Zahl 110 bezeichnet das PC-OS.

Die Zahlen 111a-111c sind PC-Prozesse von Anwendungsprogrammen, die auf dem PC-OS laufen.

Als nächstes wird der Betrieb in der Reihenfolge (1) PC-Echtzeitsteuermechanismus, (2) Treiberbetrieb, der mit Eingabe/Ausgabe-Anforderungen verbunden ist, die durch einen PC-Prozeß erzeugt werden, (3) zweite Steuervorrichtung zur Steuerung eines RT-Prozesses als eine Routine des Echtzeitsteuermechanismus, (4) zweite Steuervorrichtung, wenn ein RT-Prozeß installiert ist, der eine einzigartige Umgebung besitzt, (5) Verarbeitung der Kommunikation zwischen RT-Prozessen, (6) Verarbeitung der Kommunikation zwischen PC-Prozessen, (7) RT-Ladevorrichtungen, und (8) PC-OS-Schutzvorrichtung erörtert.

(1) Als erstes wird erörtert werden, wie der PC-Echtzeitsteuermechanismus 107 ausgeführt wird.

Aus der Sicht des PC-OS 110 ist der PC-Echtzeitsteuermechanismus 107 ein Gerätetreiber. Der Gerätetreiber wird ausgeführt entweder wenn ein Interrupt von irgendeinem Gerät anders als die CPU auftritt oder wenn eine Eingabe/Ausgabe-Anforderung von einem PC-Prozeß empfangen wird.

Fig. 2 zeigt den Fluß eines Prozesses von dem Auftreten eines Interrupts in einem normalen OS bis zur Beendigung der korrespondierenden Gerätetreiberverarbeitung.

Wenn ein Interrupt auftritt, beginnt die CPU eine Routine, die durch das PC-OS 110 für die CPU zum Zeitpunkt der OS-Initialisierung eingetragen wurde.

Die begonnene OS-Routine sichert zuerst den Systemzustand im Schritt S201, der vorliegt, wenn der Interrupt auftritt.

Als nächstes wird in Schritt 202 überprüft, ob oder ob nicht eine Dienstroutine zuvor eingetragen wurde, die dem Interrupt (Interruptdienstroutine) entspricht. Wenn keine Interruptdienstroutine bei Schritt 202 registriert ist, geht die Steuerung zu Schritt 203, bei welchem das System beim Auftreten eines unerwarteten Interrupts angehalten wird oder die Steuerung geht zu Schritt 205, bei welchem der Systemzustand wiederhergestellt wird und der Prozeß abgeschlossen wird. Zu welchem Schritt die Steuerung geht, hängt von dem Typ des PC-OS ab; dieses Thema betrifft nicht unmittelbar die Erfindung und wird daher nicht erörtert.

Wenn die Interruptdienstroutine bei Schritt 202 registriert ist, wird sie bei Schritt 204 aufgerufen. Bei Beendigung der Ausführung der Interruptdienstroutine geht die Steuerung zu Schritt 205, bei welchem der Systemzustand, der bei Schritt 201 gesichert wurde, wieder hergestellt wird und der Interruptdienst wird vollendet.

Die Interruptdienstroutine, die bei Schritt 204 aufgerufen wurde, tauscht Daten mit dem Gerät aus, das das Auftreten des Interrupts verursacht hat (wenn der Interrupt ein Dateneingabeinterrupt ist holt sie Daten von dem Gerät; wenn der Interrupt ein Datenausgabevollendungsinterrupt ist sendet sie Ausgabedaten zu dem Gerät). Im Ergebnis wird der Wartezustand des PC-Prozesses aufgehoben und die erste Steuervorrichtung 108 wird aufgerufen, um den PC-Prozeß erneut zu steuern, wenn ein PC-Prozeß, der auf eine Eingabe/Ausgabe-Anforderung wartet, existiert.

Wenn die Steuervorrichtung direkt von einer Interruptdienstroutine in einem normalen OS gerufen wird, tritt ein Widerspruch in der Systemverarbeitung auf. Daher wird ein Mechanismus zur Verfügung gestellt, um die erste Steuervorrichtung 108 nach der Vervollendung des Interruptdienstes aufzurufen.

(2) Als nächstes wird aufgrund eines Flußdiagrammes der Fig. 3 ein Gerätetreiber-Ausführungsprozeß erörtert werden, der mit einer Eingabe/Ausgabe-Anforderung verbunden ist, die durch einen PC-Prozeß 111 erzeugt wurde.

Wenn eine Eingabe/Ausgabe-Anforderung von einem PC-Prozeß ausgegeben wird, wird der Treiber-Einbau-Mechanismus, ein Teil des PC-OS 110, aufgerufen. Der Treiber-Einbau-Mechanismus beginnt bei Schritt 301 eine Routine zur Annahme der Gerätetreiberanforderung, die im Typ der Eingabe/Ausgabe-Anforderung von dem PC-Prozeß entspricht. Beispielsweise sind die Routinen zur Annahme der Anforderung eingeteilt in diejenigen für Daten lesen, Daten schreiben, direkte Gerätesteuerung, etc. und sie werden bei der Verarbeitung der Initialisierung der Gerätetreiber in dem Treiber-Einbau-Mechanismus registriert. Die Routine zur Annahme der Gerätetreiberanforderung tauscht Daten mit einem vorgegebenen Gerät als Antwort auf den Anforderungsinhalt aus. Da das Gerät verglichen mit der CPU-Geschwindigkeit sehr langsam ist, versetzt der Treiber-Einbau-Mechanismus in Schritt 302 den PC-Prozeß in einen Eingabe/Ausgabe-Vollendungswartezustand.

Da der PC-Prozeß dennoch bei Schritt 302 in den Wartezustand eintritt, wird bei Schritt 303 die erste Steuervorrichtung gerufen, um den nächsten auszuführenden PC-Prozeß auszuwählen und der Prozeß zur Eingabe/Ausgabe-Anforderung wird vollendet.

Der PC-Prozeß, der in den Eingabe/Ausgabe-Vollen-

dungswartezustand versetzt wurde, wird aus dem Wartezustand durch die in Fig. 2 gezeigte Interruptdienstroutine entlassen.

Wie wir erörtert haben, wird die erste Steuervorrichtung 108, die in dem PC-OS enthalten ist, während der Gerätetreiberverarbeitung nicht ausgeführt. Im Ergebnis wird der Gerätetreiber verarbeitet, wobei er Vorrang über die PC-Prozesse 111a—111c und die erste Steuervorrichtung 108 erhält.

(3) Als nächstes wird der Betrieb der zweiten Steuervorrichtung 101 zur Steuerung der Ausführung eines RT-Prozesses unter Bezug auf Fig. 4 erörtert werden.

Fig. 4 ist ein Flußdiagramm, um die Verarbeitung der zweiten Steuervorrichtung 101 zu zeigen, wenn ein RT-Prozeß als eine Routine des PC-Echtzeitkontrollmechanismus installiert wird.

Wenn die zweite Steuervorrichtung 101 aus Schritt 204 in Fig. 2 aufgerufen wird, führt sie zuerst Schritt 401 aus. Die zweite Steuervorrichtung 101 überprüft ein Kennzeichen, das anzeigt, daß der PC-Echtzeitsteuermechanismus ausgeführt wird. Wenn das Kennzeichen AN ist, führt die zweite Steuervorrichtung 101 keine Operation durch, beendet den Prozeß und kehrt zu Schritt 205 in Fig. 2 zurück.

Wenn das Kennzeichen bei Schritt 401 AUS ist, setzt die zweite Steuervorrichtung 101 das Kennzeichen im Schritt 402 auf AN und geht zu Schritt 403 und überprüft, ob oder ob nicht ein ausführbarer RT-Prozeß existiert. Um dies zu tun werden die Anfangsadressen von RT-Prozessen (in der Ausführungsform Anfangsadressen von Routinen) in einer Tabelle festgehalten und es kann bestimmt werden, daß alle RT-Prozesse, die in der Tabelle festgehalten werden, in einem ausführbaren Zustand sind. Wenn ein Interrupt von einem Gerät regelmäßig auftritt wird die Tabelle mit einem Zähler versehen, wobei die Operation auch bei unterschiedlichen periodischen Zeiten für jeden RT-Prozeß begonnen werden kann.

Wenn bei Schritt 403 ein ausführbarer RT-Prozeß existiert, geht die zweite Steuervorrichtung 101 zu Schritt 404 und ruft die in der Tabelle eingetragene Routine auf. Die aufgerufene Routine (RT-Prozeß) führt eine Verarbeitung gemäß des RT-Prozesses durch und gibt dann die Steuerung an Schritt 404 zurück, woraufhin die Steuerung zu Schritt 403 geht. Schritt 403 und 404 werden wiederholt ausgeführt, wobei alle ausführbaren RT-Prozesse die in der Tabelle eingetragen sind, ausgeführt werden. Dann geht die zweite Steuervorrichtung 101 nach Schritt 405, setzt das Kennzeichen auf 0 (AUS) und beendet den Prozeß.

In diesem Falle wird der RT-Prozeß bei der Rückkehr von der Routine wie in Fig. 4 gezeigt beendet.

(4) Als nächstes wird aufgrund eines Flußdiagrammes in Fig. 5 der Betrieb der zweiten Steuervorrichtung 101 erörtert, wenn ein RT-Prozeß, der eine einzigartige Umgebung besitzt, installiert wird statt als eine Routine des PC-Echtzeitsteuermechanismus.

In diesem Falle besteht der RC-Prozeß aus Initialisierungsverarbeitung des RT-Prozesses selbst und wiederholter Verarbeitung gemäß dem Prozeß wie in Fig. 5 gezeigt. Die wiederholte Verarbeitung wird beendet durch den Aufruf von Achtung Warten zur Aufgabe des Ausführungsrechtes.

Wenn die zweite Steuervorrichtung 101 von Schritt 204 in Fig. 2 aufgerufen wird, führt sie zuerst die Schritte 401 und 402 in Fig. 5 wie diejenigen in Fig. 4 aus.

Als nächstes wird in Schritt 403 die zweite Steuervorrichtung 101, ob oder ob nicht ein ausführbarer RT-Pro-

zeß existiert, auf dieselbe Weise wie unter Bezug auf das Flußdiagramm in Fig. 4 beschrieben. Wenn in Schritt 403 ein ausführbarer RT-Prozeß existiert, geht die zweite Steuervorrichtung 101 zu Schritt 501 und stellt die Zustandsinformation des RT-Prozesses wieder her, den sie bei Schritt 403 gefunden hat. Wenn die Zustandsinformation des RT-Prozesses wiederhergestellt ist, wird der RT-Prozeß beginnend mit Schritt 503 wieder ausgeführt. Die Verarbeitung des RT-Prozesses wird hier erörtert.

Wenn der RT-Prozeß zum ersten Mal aufgerufen wird, führt er seine eigene Initialisierungsverarbeitung bei Schritt 502 durch und geht zu Schritt 503. Um den RT-Prozeß zum ersten Mal aufzurufen, kann er in einer Initialisierungsverarbeitung des PC-Echtzeitsteuermechanismus des Gerätetreibers aufgerufen werden, oder er kann in einer Gerätestartanforderung aufgerufen werden, die von einem PC-Prozeß ausgegeben wird.

In Schritt 503 wird eine Verarbeitung gemäß dem RT-Prozeß durchgeführt und dann wird in Schritt 504 Warten aufgerufen.

Der Warte-Aufruf ist eine Funktion, die von dem PC-Echtzeitsteuermechanismus für den RT-Prozeß zur Verfügung gestellt wird und benutzt wird, um zeitweise die Ausführung des RT-Prozesses anzuhalten. Diese Funktion ist für die Erfindung nicht unentbehrlich, aber sie erleichtert die Beschreibung der Ausführungsform der Erfindung. Der aufgerufene RT-Prozeß wird im Schritt 505 außer den ausführbaren Zustand gesetzt und die Zustandsinformation des RT-Prozesses wird in Schritt 506 festgehalten, daraufhin wird im Schritt 403 die zweite Steuervorrichtung 101 wieder aufgerufen.

Um die Zustandsinformation des RT-Prozesses in Schritt 501 wieder herzustellen, wird die Zustandsinformation, die in Schritt 506 festgehalten wurde, wiederhergestellt und der RT-Prozeß beginnt wieder mit der Ausführung in Schritt 503.

Wenn die Schritte 403 und 501 bis 506 wiederholt werden und alle ausführbaren RT-Prozesse außer den ausführbaren Zustand durch Aufruf von Warten gesetzt sind, wird in Schritt 403 festgestellt, daß keine ausführbaren RT-Prozesse existieren. In Schritt 405 wird das Kennzeichen auf AUS gesetzt und die Steuerung kehrt zu dem PC-OS Interruptdienst zurück (Schritt 204 in dem Flußdiagramm in Fig. 2).

(5) Als nächstes wird die Verarbeitung der Vorrichtung zur Kommunikation zwischen RT-Prozessen erörtert.

Die RT-Prozesse werden als unabhängige Verarbeitungen unter Bezug auf die Flußdiagramme in den Fig. 4 und 5 beschrieben, allgemeine Prozesse jedoch führen eine Verarbeitung durch, während sie die Verarbeitung synchronisieren und Daten miteinander austauschen.

Ein Prozeßablauf wird erörtert werden, indem der einfachste Datenaustausch zwischen zwei RT-Prozessen 102a und 102b als ein Beispiel genommen wird.

Fig. 6 ist ein Flußdiagramm, das den Datenempfang des RT-Prozesses 102a und die Datenübertragung des RT-Prozesses 102b anzeigt.

Wenn der RT-Prozeß 102a im Schritt 601 Daten empfängt, wird zuerst die Vorrichtung zur Kommunikation zwischen RT-Prozessen in Schritt 602 aufgerufen und dieser überprüft ob oder ob nicht Daten existieren. Wenn Daten existieren kopiert die Vorrichtung zur Kommunikation zwischen RT-Prozessen die Daten in einem Speicherbereich, der bestimmt wird, wenn im Schritt 603 ein Datenempfang des RT-Prozesses aufgerufen wird, sie beendet den Datenempfangsprozess, ruft

Warten in Schritt 504 des RT-Prozesses auf und verläßt den ausführbaren Zustand.

Wenn keine Daten im Schritt 602 existieren, setzt die Vorrichtung zur Kommunikation zwischen RT-Prozessen den RT-Prozeß 102a außer den ausführbaren Zustand in Schritt 604, hält die RT-Prozeßzustandsinformation in Schritt 605 fest und ruft die zweite Steuervorrichtung 101 in Schritt 606. Die aufgerufene zweite Steuervorrichtung 101 ist dieselbe wie diejenige, die bei der Warten-Verarbeitung beschrieben wurde (Schritt 403 in Fig. 5).

Wenn der RT-Prozeß 102b ausgewählt wird und die Zustandsinformation als Ergebnis der Verarbeitung der zweiten Steuervorrichtung 101 wiederhergestellt wird, so wird die Ausführung des RT-Prozesses 102b unmittelbar nach Warten erneut begonnen. Der RT-Prozeß 102b geht zu Schritt 607 und ruft die Datenübertragung auf. Wenn die Datenübertragung aufgerufen wird, wird die Vorrichtung zur Kommunikation zwischen RT-Prozessen in Schritt 608 ausgeführt und sie prüft, ob oder ob nicht ein Prozeß in einem Datenwartezustand existiert.

Wenn kein RT-Prozeß in einem Datenwartezustand existiert, kopiert die Vorrichtung zur Kommunikation zwischen RT-Prozessen in Schritt 609 die Daten in einen Speicherbereich, der in der Vorrichtung zur Kommunikation zwischen RT-Prozessen enthalten ist, vollendet die Datenübertragung, kehrt zu dem RT-Prozeß 102b zurück und verläßt den ausführbaren Zustand durch Aufruf von Warten. In der Beschreibung tritt der RT-Prozeß 102a im Schritt 604 in den Datenwartezustand ein und folglich wird im Schritt 608 festgestellt, daß ein Prozeß existiert, der auf Daten wartet und die Vorrichtung zur Kommunikation zwischen RT-Prozessen geht zu Schritt 610.

In Schritt 610 kopiert die Vorrichtung zur Kommunikation zwischen RT-Prozessen die Daten in den Speicherbereich, der durch den RT-Prozeß bestimmt wird und versetzt in Schritt 611 den RT-Prozeß in den ausführbaren Zustand, wobei die Verarbeitung der Datenübertragung vollständig ist, die Steuerung zu dem RT-Prozeß 102b zurückkehrt, in Schritt 504b Warten aufgerufen wird und der RT-Prozeß 102b den ausführbaren Zustand verläßt.

Wenn der RT-Prozeß 102a als Prozeß in ausführbarem Zustand in Schritt 403 in dem Flußdiagramm in Fig. 5 ausgewählt wird und der Zustand in Schritt 501 wieder hergestellt wird, dann kehrt die Steuerung zu Schritt 606 zurück.

Die Datenempfangsverarbeitung des RT-Prozesses 102a ist dann vollständig. Wenn im Schritt 504a Warten aufgerufen wird, verläßt der RT-Prozeß 102a den ausführbaren Zustand und die zweite Steuervorrichtung 101 gibt die Steuerung an den PC-OS-Interruptdienst zurück, weil kein Prozeß in einem ausführbaren Zustand existiert.

(6) Als nächstes wird die Verarbeitung der Vorrichtung zur Kommunikation zwischen PC-Prozessen unter Bezug auf Fig. 7 und 8 erörtert werden.

In einem normalen Computerbetriebssystem wie PC-OS, können PC-Prozesse oder RT-Prozesse die Verarbeitung untereinander synchronisieren oder Daten miteinander austauschen. Wenn PC- und RT-Prozesse gemischt sind, wird die Synchronisation des Datenaustausches zwischen den PC- und RT-Prozessen notwendig.

Die Prozeßabläufe werden erörtert, indem der einfachste Datenaustausch zwischen PC- und RT-Prozessen als Beispiel genommen wird. Fig. 7 ist ein Flußdiagramm

gramm für den Empfang von Daten durch den PC-Prozeß und die Übertragung von Daten durch den RT-Prozeß. Fig. 8 ist ein Flußdiagramm für die Übertragung von Daten durch den PC-Prozeß und den Empfang von Daten durch den RT-Prozeß.

Tatsächlich entspricht die Datenübertragung in dem PC-Prozeß einer Datenausgabeaufforderung, die an den Gerätetreiber ausgegeben wird, der dem PC-Echtzeitsteuermechanismus gleichwertig ist und der Datenempfang entspricht einer Dateneingabeaufforderung, die an diesen ausgegeben wird. Die Funktion kann in dieser Form für den PC-Prozeß zur Verfügung gestellt werden, oder eine Bibliothek kann für den PC-Prozeß zur Verfügung gestellt werden, zur Verknüpfung einer Datenausgabeaufforderung mit einer Datenübertragung und einer Dateneingabeaufforderung mit einem Datenempfang.

Zuerst wird unter Bezug auf Fig. 7 der Betrieb zum Empfang von Daten durch den PC-Prozeß und zur Übertragung von Daten durch den RT-Prozeß beschrieben werden.

Wenn die erste Steuervorrichtung 108 den PC-Prozeß ausführt und einen Datenempfang in Schritt 701 anfordert, ruft der Treiber-Einbau-Mechanismus die Vorrichtung zur Kommunikation zwischen PC-Prozessen des PC-Echtzeitsteuermechanismus als eine Dateneingabeaufforderung auf.

Die Vorrichtung zur Kommunikation zwischen PC-Prozessen überprüft als erstes den Schritt 702, ob oder ob nicht Daten existieren. Wenn Daten existieren, geht die Vorrichtung zur Kommunikation zwischen PC-Prozessen zu Schritt 703 und gibt die Steuerung an den Treiber-Einbau-Mechanismus als Dateneingabeaufforderungsvollendung wie der normale Gerätetreiber zurück. Im Ergebnis wird der PC-Prozeß ausgeführt, wobei Schritt 701 als Datenempfangsvollendung fortgesetzt wird.

Wenn in Schritt 702 keine Daten existieren, geht die Vorrichtung zur Kommunikation zwischen PC-Prozessen zu Schritt 704 und setzt den PC-Prozeß in einen Wartezustand auf Dateneingabe von einem Gerät und benachrichtigt dann das PC-OS.

Als nächstes geht die Vorrichtung zur Kommunikation zwischen PC-Prozessen zu Schritt 705 und gibt die Steuerung an den Treiber-Einbau-Mechanismus 109 als Dateneingabeaufforderungsvollendung zurück. Da der PC-Prozeß in den Geräteeingabewartezustand versetzt ist, ruft der Treiber-Einbau-Mechanismus 1 (109) die erste Steuervorrichtung 108 in das PC-OS auf zur Ausführung eines anderen PC-Prozesses als des PC-Prozesses im Geräteeingabewartezustand.

Wenn die zweite Steuervorrichtung 101 als ein Geräteinterrupt für das PC-OS aufgerufen wird und folglich der RT-Prozeß ausgeführt wird, wird die Vorrichtung zur Kommunikation zwischen PC-Prozessen in Schritt 706 aufgerufen und überprüft, ob oder ob nicht ein Prozeß existiert, der auf einen Datenempfang wartet.

Wenn in Schritt 706 kein Prozeß existiert, der auf einen Datenempfang wartet, hält die Vorrichtung zur Kommunikation zwischen PC-Prozessen die Daten fest und kehrt zu dem RT-Prozeß zurück. Bei dem Beispiel wartet der PC-Prozeß auf einen Datenempfang und folglich geht die Vorrichtung zur Kommunikation zwischen PC-Prozessen zu Schritt 708 und kopiert Daten in den PC-Prozeßspeicherbereich, der festgelegt wird, wenn der PC-Prozeß die Dateneingabeaufforderung erzeugt. Daraufhin geht sie zu Schritt 709 und benachrichtigt das PC-OS, daß der Dateneingabewartezustand des

PC-Prozesses aufgegeben wurde.

Das PC-OS ändert den Zustand des PC-Prozesses von dem Eingabewartezustand in einen ausführbaren Zustand. Wie zuvor unter Bezug auf das Flußdiagramm in Fig. 2 beschrieben wird jedoch erkannt, daß ein Geräteinterruptservice von dem PC-OS ausgeführt wird, und folglich wird die erste Steuervorrichtung 108 nicht aufgerufen und die Steuerung wird an Schritt 709 zurückgegeben, wobei die Verarbeitung der Datenübertragung der Vorrichtung zur Kommunikation zwischen PC-Prozessen beendet wird und die Steuerung an den RT-Prozeß zurückgegeben wird.

Wenn alle ausführbaren RT-Prozesse abgearbeitet sind, wird die Steuerung von der zweiten Steuervorrichtung 101 an den Interruptdienst des PC-OS zurückgegeben. Wie zuvor beschrieben, unter Bezug auf das Flußdiagramm in Fig. 2, existiert bei dem Interruptservice der PC-Prozeß, dessen Geräteeingabewartezustand aufgelöst wurde, und daher wird die erste Steuervorrichtung 108 des PC-OS gerufen und die Ausführung des PC-Prozesses kann in manchen Fällen in der Form einer Datenempfangsvollendung gestartet werden.

Als nächstes wird unter Bezug auf Fig. 8 der Betrieb erörtert werden, bei dem der PC-Prozeß Daten überträgt und der RT-Prozeß Daten empfängt.

Wenn die zweite Steuervorrichtung 101 von einem Geräteinterrupt aufgerufen wird und in der Folge der RT-Prozeß ausgeführt und Datenempfang aufgerufen wird, dann wird die Vorrichtung zur Kommunikation zwischen PC-Prozessen aufgerufen.

Die Vorrichtung zur Kommunikation zwischen PC-Prozessen überprüft als erstes im Schritt 801, ob oder ob nicht Daten existieren. Wenn Daten existieren, kopiert die Vorrichtung zur Kommunikation zwischen PC-Prozessen die Daten in den Speicherbereich, der durch den RT-Prozeß in Schritt 802 bestimmt wurde, und beendet den Datenempfang und gibt dann die Steuerung an den RT-Prozeß zurück.

Wenn in Schritt 801 keine Daten existieren, so versetzt die Vorrichtung zur Kommunikation zwischen PC-Prozessen den RT-Prozeß außer einen ausführbaren Zustand und den Prozeß in einen Datenempfangwartezustand in Schritt 803 und hält die Zustandsinformation in Schritt 804 fest, ruft dann die zweite Steuervorrichtung 101 in Schritt 805. Diese aufgerufene zweite Steuervorrichtung entspricht Schritt 403 in Fig. 5, wie bei der Beschreibung von Warten.

Wenn alle ausführbaren RT-Prozesse außer dem RT-Prozeß, der in den Datenempfangwartezustand versetzt ist, ausgeführt sind, gibt die zweite Steuervorrichtung 101 die Steuerung an den Interruptservice des PC-OS zurück und der Geräteinterruptservice ist vollendet. Folglich wird die erste Steuervorrichtung 108 des PC-OS aufgerufen.

Bei Abschluß des Geräteinterruptservices (wenn keiner der Prozesse im Geräteeingabewartezustand aus dem Wartezustand in der Gerätetreiberverarbeitung entlassen wurde) werden einige Betriebssysteme zu dem Punkt des Auftretens des Interrupts zurückkehren. In vielen Fällen wird irgendein PC-Prozeß ausgeführt, und die Verarbeitung wird bei Abschluß des Interruptservices wieder begonnen. Da das PC-OS die CPU den PC-Prozessen gleichmäßig zuweist, werden die PC-Prozesse früher oder später durch die erste Steuervorrichtung 108 ausgeführt.

Wenn der PC-Prozeß eine Anforderung zur Datenübertragung in Schritt 806 erzeugt, ruft der Treiber-Einbau-Mechanismus die Vorrichtung zur Kommunikation

zwischen PC-Prozessen des PC-Echtzeitsteuermechanismus als eine Datenausgabeanforderung auf. Die Vorrichtung zur Kommunikation zwischen PC-Prozessen überprüft zuerst in Schritt 807, ob oder ob nicht ein RT-Prozeß in einem Datenwartezustand existiert. Wenn ein RT-Prozeß in einem Datenwartezustand nicht existiert, kopiert die Vorrichtung zur Kommunikation zwischen PC-Prozessen in Schritt 808 Daten in den Speicherbereich der Vorrichtung zur Kommunikation zwischen PC-Prozessen.

Als nächstes benachrichtigt im Schritt 809 die Vorrichtung zur Kommunikation zwischen PC-Prozessen den Treiber-Einbau-Mechanismus von dem Abschluß der Datenausgabeanforderung und beendet die Verarbeitung. Folglich wird die Ausführung des PC-Prozesses neu begonnen als Datenübertragungsabschluß. In dem Beispiel wird der RT-Prozeß in den Datenwartezustand versetzt, die Vorrichtung zur Kommunikation zwischen PC-Prozessen geht von Schritt 807 zu Schritt 810 und kopiert die Daten in den Speicherbereich, der durch den RT-Prozeß bestimmt ist.

Als nächstes versetzt in Schritt 811 die Vorrichtung zur Kommunikation zwischen PC-Prozessen den RT-Prozeß in einen ausführbaren Zustand. Dann geht sie zum Schritt 809 und benachrichtigt den Treiber-Einbau-Mechanismus vom Abschluß der Datenausgabeanforderung und vollendet die Verarbeitung. Der RT-Prozeß, der im Schritt 811 in den ausführbaren Zustand versetzt wurde, wird durch die zweite Steuervorrichtung 101 ausgeführt, wenn ein anderer Gerätetreiberinterrupt auftritt, und die Datenempfangsverarbeitung des RT-Prozesses ist abgeschlossen.

Nachdem der RT-Prozeß im Schritt 811 in den ausführbaren Zustand versetzt wurde, wird der Treiber-Einbau-Mechanismus von dem Abschluß der Datenausgabeanforderung benachrichtigt. Um die Ansprechbarkeit auf den RT-Prozeß zu erhöhen, kann jedoch die zweite Steuervorrichtung 101 aufgerufen werden, um den RT-Prozeß auszuführen, bevor der Treiber-Einbau-Mechanismus von dem Abschluß der Datenausgabeanforderung benachrichtigt wird. In diesem Falle muß der Treiber-Einbau-Mechanismus von dem Abschluß der Datenausgabeanforderung in Warten benachrichtigt werden, das von dem RT-Prozeß aufgerufen wird, bevor die Verarbeitung beendet wird.

(7) Als nächstes wird die Verarbeitung der RT-Ladevorrichtung 105 erörtert werden.

In allgemeinen Computersystemen müssen Gerätetreiber verglichen mit der Entwicklung von Anwendungen sorgfältig entwickelt werden und sind schwer zu entwickeln. Da der PC-Echtzeitsteuermechanismus der Erfindung RT-Prozesse als interne Routinen des Gerätetreibers ausführt, verursacht die Entwicklung verglichen mit RT-Prozessen, die früher als RT-OS-Anwendungen entwickelt wurden, Schwierigkeiten.

Daher ist die RT-Ladevorrichtung 105 ein Mechanismus, um eine Anwendung als ein RT-Prozeß 102 des PC-Echtzeitsteuermechanismus des Gerätetreibers aufzunehmen, die als ein PC-Prozeß 111c entwickelt und getestet wurde.

Die RT-Ladevorrichtung 105 hängt stark von den Begrenzungen ab, die aufgestellt wurden, falls ein RT-Prozeß als PC-Prozeß hergestellt und getestet wurde. Wenn beispielsweise die RT-Ladevorrichtung eine Registrierungsanforderung eines RT-Prozesses als einen RT-Prozeß ausführt, der als eine Routine eines PC-Prozesses hergestellt wurde, kann sie den Prozeß (Routine) in der Größe einer Routine, die an der vorbestimmten

Startadresse für Routinen beginnt, in einen Speicherbereich in dem PC-Echtzeitsteuermechanismus des Gerätetreibers kopieren und die Startadresse der Routine in der Tabelle eintragen, die zuvor unter Bezug auf das Flußdiagramm in Fig. 4 beschrieben wurde.

Um beispielsweise den gesamten PC-Prozeß zu registrieren, der als vom PC-Prozeß erzeugt und getestet wurde, kann die RT-Ladevorrichtung einen PC-Prozeß, der auf einer externen Speichereinheit gespeichert ist, in einen Speicherbereich in dem PC-Echtzeitsteuermechanismus lesen und die Anfangsadresse in der Tabelle registrieren, die zuvor unter Bezug auf das Flußdiagramm in Fig. 4 beschrieben wurde.

Wenn die Routine oder PC-Prozeß, der als ein RT-Prozeß erzeugt und getestet wurde, Funktionen benutzt, die durch das PC-OS zur Verfügung gestellt werden, beispielsweise einen Systemaufruf, dann muß selbstverständlich die Ladevorrichtung dieselbe Funktion in dem PC-Echtzeitsteuermechanismus zur Verfügung stellen. Da die Funktionen als einfache Unterrouinen des PC-Prozesses programmiert sind, müssen die Adressen der Unterrouinen selbstverständlich aufgelöst werden, während die Unterrouinen als RT-Prozesse in den PC-Echtzeitsteuermechanismus geladen oder ausgeführt werden.

Als ein Verfahren zur Auflösung der Adressen können beispielsweise nach der Beendigung des Tests des PC-Prozesses nur die Unterrouinen in einen Zustand unaufgelöster Adressen weggkompiliert werden und die RT-Ladevorrichtung kann während des Kopierens des Programmes unaufgelöste Adressen mit den korrespondierenden Unterrouinen in dem PC-Echtzeitsteuermechanismus ersetzen.

Weiterhin stehen mehrere bekannte Verfahren zur Verfügung; jedes Verfahren kann in der Erfindung verwendet werden. Die Funktionen, die durch das PC-OS zur Verfügung gestellt werden, können als eine Nutzungsbegrenzung abgeschaltet werden, die aufgestellt wird, um PC-Prozesse als RT-Prozesse zu erzeugen.

(8) Als nächstes wird die PC-OS-Schutzvorrichtung 106 erörtert werden.

Das PC-OS, das den Treibereinbaumechanismus besitzt, stellt PC-OS-Funktionen zur Verfügung, die von dem Gerätetreiber verwendet werden können. Diese Funktionen unterscheiden sich normalerweise von den Funktionen, die durch das PC-OS für PC-Prozesse zur Verfügung gestellt werden. Der Gerätetreiber läuft häufig als ein Teil des PC-OS und leichtfertige Benutzung der Funktion, die für den Gerätetreiber an einem RT-Prozeß zur Verfügung gestellt wird, kann ein ernsthaftes Problem verursachen, das zu einem Anhalten des Systems führt.

Die PC-OS-Schutzvorrichtung ist eine Konversionsroutine, um die RT-Prozesse sicherer mit den Funktionen, die durch das PC-OS für den Gerätetreiber zur Verfügung gestellt werden, zu versorgen und sie führt eine strenge Fehlerüberprüfung durch, überprüft die Zugangsadresse, schließt Verarbeitungen aus, die einen Widerspruch für den PC-OS-Betrieb verursachen, etc.

Ausführungsform 2

Eine zweite Ausführungsform der Erfindung wird unter Bezug auf die Fig. 9 bis 12 erörtert.

In Fig. 9 bezeichnet die Zahl 901 die Vorrichtung zur Überwachung von Nutzungsrechten zur Überwachung der Belegzeit einer CPU in einem PC-Echtzeitsteuermechanismus. Der PC-Echtzeitsteuermechanismus, der als

ein Gerätetreiber läuft, wird nicht durch die erste Steuervorrichtung gesteuert. Wenn eine große Anzahl von RT-Prozessen läuft, wird daher die Steuerung nicht an die erste Steuervorrichtung übertragen und es tritt ein Zustand auf, bei welchem ein PC-Prozeß nicht längere Zeit laufen kann.

Die Vorrichtung zur Überwachung von Nutzungsrechten überwacht und zählt die Zeit, während der der PC-Echtzeitsteuermechanismus die CPU besetzt und gibt zwangsweise die Steuerung an das PC-OS unter einer geeigneten Bedingung zurück, so daß die CPU ordnungsgemäß zwischen den PC-Echtzeitsteuermechanismus und der CPU verteilt werden kann.

Die Ziffern 101-111 in Fig. 9 sind dieselben wie die in Fig. 1 gezeigten.

Als nächstes wird der Betrieb erörtert werden.

Die Vorrichtung zur Überwachung von Nutzungsrechten läuft als eine Interruptdienstroutine, die von dem Schritt 204 in Fig. 2 aufgerufen wird.

Im Schritt 1001 bestimmt die Vorrichtung zur Überwachung von Nutzungsrechten zuerst, ob oder ob nicht der PC-Echtzeitkontrollmechanismus in Betrieb ist. Das Bestimmungsverfahren kann ähnlich dem Verfahren in Schritt 401 in Fig. 4 sein.

Wenn der PC-Echtzeitsteuermechanismus in Betrieb ist, zählt die Vorrichtung zur Überwachung von Nutzungsrechten die Betriebszeit des PC-Echtzeitsteuermechanismus. Das Zählverfahren kann einen einfachen Zähler verwenden oder die Betriebszeit aufsummieren.

Als nächstes bestimmt im Schritt 1003 die Vorrichtung zur Überwachung von Nutzungsrechten, ob oder ob nicht die Zählinformation, wie sie in Schritt 1002 zur Verfügung gestellt wird, eine vorherbestimmte Bedingung erfüllt. Die vorherbestimmte Bedingung kann der obere Grenzwert der kontinuierlichen CPU-Belegzeit des PC-Echtzeitsteuermechanismus sein, oder sie kann beispielsweise der obere Grenzwert des Prozentanteils der CPU-Belegung innerhalb einer gegebenen Zeit sein.

Wenn in Schritt 1003 festgestellt wird, daß die Bedingung erfüllt ist, d. h. der Zustand ein Zustand ist, in welchem das CPU Nutzungsrecht zwangsweise von dem PC-Echtzeitsteuermechanismus an das PC-OS zurückgegeben wird, geht die Vorrichtung zur Überwachung von Nutzungsrechten zu Schritt 1004, setzt ein Zwangsbeendigungskennzeichen und beendet die Verarbeitung.

Dieses Zwangsbeendigungskennzeichen wird durch eine zweite Steuervorrichtung überprüft. Fig. 11 zeigt einen Verarbeitungsablauf der zweiten Steuervorrichtung, welchem der Überprüfungsprozeß hinzugefügt wurde.

Fig. 11 unterscheidet sich von Fig. 4 lediglich darin, daß Schritt 404 in Fig. 4 von einem zusätzlichen Schritt 1101 gefolgt wird, um zu überprüfen, ob oder ob nicht das Zwangsbeendigungskennzeichen gesetzt ist.

Wenn in Schritt 1101 das Zwangsbeendigungskennzeichen gesetzt ist, wird die Verteilung der CPU an den RT-Prozeß beendet und die Steuerung wird an das PC-OS zurückgegeben, selbst wenn ein RT-Prozeß in einem ausführbaren Zustand existiert.

Wenn die Bedingung in Schritt 1003 nicht erfüllt ist, wird als nächstes die Verarbeitung der Vorrichtung zur Überwachung von Nutzungsrechten beendet, da der PC-Echtzeitsteuermechanismus in Betrieb ist und die RT-Prozeßverarbeitung wird an dem Punkt des Auftretes des Interrupts wieder aufgenommen.

Wenn in Schritt 1001 festgestellt wird, daß der PC-Echtzeitsteuermechanismus nicht ausgeführt wird, geht

die Steuerung zu Schritt 1005, bei welchem gemäß der zurückliegenden Zählinformation wie in Schritt 1003 überprüft wird, ob oder ob nicht die Bedingung erfüllt ist. Schritt 1005 wird wirksam, wenn der obere Grenzwert des Prozentanteils an CPU-Belegung innerhalb einer gegebenen Zeit überschritten wird.

Wenn in Schritt 1005 festgestellt wird, daß die Bedingung erfüllt ist, wird die Vorrichtung zur Überwachung von Nutzungsrechten angehalten und die Steuerung wird an das PC-OS zurückgegeben.

Wenn in Schritt 1005 festgestellt wird, daß die Bedingung nicht erfüllt ist, d. h. daß der PC-Echtzeitsteuermechanismus in Betrieb ist, geht die Steuerung zum Schritt 1006 und die zweite Steuervorrichtung wird aufgerufen. Bei Beendigung der Verarbeitung aller ausführbarer RT-Prozesse überträgt die zweite Steuervorrichtung die Kontrolle an Schritt 1007, bei welchem die Vorrichtung zur Überwachung von Nutzungsrechten die Betriebsdauer des PC-Echtzeitsteuermechanismus selbst und dann die Steuerung an das PC-OS zurückgibt.

Als nächstes wird ein weiteres Verarbeitungsbeispiel für zwangsweise Rückgabe der CPU-Nutzungsrechte an das PC-OS entsprechend Fig. 12 erörtert.

In der Figur sind die Schritte 1001—1007 dieselben wie diejenigen in dem Flußdiagramm in Fig. 10.

Wenn bei Schritt 1003 festgestellt wird, daß die Bedingung erfüllt ist geht die Vorrichtung zur Überwachung von Nutzungsrechten zu Schritt 1201 und versetzt alle augenblicklich in einem ausführbaren Zustand befindlichen RT-Prozesse außerhalb des ausführbaren Zustandes und beendet dann die Verarbeitung.

Wenn in Schritt 1005 festgestellt wird, daß die Bedingung nicht erfüllt ist, stellt die Vorrichtung zur Überwachung von Nutzungsrechten alle RT-Prozesse wieder her, die in Schritt 1201 angehalten wurden, in den ausführbaren Zustand in Schritt 1203, wieder her und ruft dann im Schritt 1006 die zweite Steuervorrichtung auf.

Ausführungsform 3

Als nächstes wird unter Bezug auf die Fig. 13 und 14 eine dritte Ausführungsform der Erfindung erörtert werden.

In Fig. 13 sind die Ziffern 1301a und 1301b Vorrichtungen zur Konkurrenzkontrolle zur Ausübung einer ausschließlichen Steuerung, wenn RT-Prozesse Datenaustausch durchführen und zur Synchronisierungsverarbeitung über PC-Echtzeitsteuermechanismen.

Die weiteren Elemente sind dieselben wie diejenigen in Fig. 1.

Als nächstes wird der Betrieb entsprechend einem Flußdiagramm in Fig. 14 erörtert werden.

Die Vorrichtung zur Konkurrenzkontrolle in dieser Ausführungsform stellt einen Zugangspunkt zur Verfügung, um Funktionen für Kommunikation zwischen RT-Prozessen, Synchronisierung etc. für einen anderen PC-Echtzeitsteuermechanismus zur Verfügung zu stellen. Welche Funktionen, wie Kommunikationen zwischen Prozessen und Synchronisierung, als PC-Echtzeitsteuermechanismus zur Verfügung gestellt werden, kann nach Wunsch bestimmt werden und ist nicht direkt verknüpft mit der Erfindung und wird daher hier nicht erörtert werden.

In Schritt 1401 überprüft zuerst die Vorrichtung zur Konkurrenzkontrolle, ob eine Verarbeitungsanfrage, die an den Zugangspunkt der Vorrichtung zur Konkurrenzkontrolle gegeben wurde, eine Anfrage ist, die von dem PC-Echtzeitsteuermechanismus ausgegeben wurde.

de, zu welchem die Vorrichtung zur Konkurrenzkontrolle gehört, zu einem anderen PC-Echtzeitsteuermechanismus oder eine Anfrage, die von einem anderen PC-Echtzeitsteuermechanismus ausgegeben wurde. Um diese Kontrolle durchzuführen, kann ein Code, der fähig ist anzuzeigen, welcher PC-Echtzeitsteuermechanismus die Verarbeitung anfragt, in die Verarbeitungsanfragedaten eingefügt werden oder eine besondere Zahl kann jedem PC-Echtzeitsteuermechanismus zugeordnet und in einer besonderen Position, beispielsweise in den Verarbeitungsanfragedaten eingefügt werden.

Wenn die Verarbeitungsanforderung nicht von einem anderen PC-Echtzeitsteuermechanismus im Schritt 1401 ausgegeben wurde, geht die Vorrichtung zur Konkurrenzkontrolle zu Schritt 1402 und sendet die Verarbeitungsanfragedaten an den Zugangspunkt der Vorrichtung zur Konkurrenzsteuerung des anderen PC-Echtzeitsteuermechanismus (1301a in Fig. 13).

Wenn die Verarbeitungsanfrage von einem anderen PC-Echtzeitsteuermechanismus in Schritt 1401 ausgegeben wurde, geht die Vorrichtung zur Konkurrenzkontrolle zu Schritt 1403 und ruft eine Routine auf zur Ausführung der angefragten Verarbeitung.

Wie wir erörtert haben, werden gemäß der Erfindung Anwendungen und Echtzeitprozesse basierend auf dem Betriebssystem betrieben, das auf einer CPU läuft, daher kann die Verbesserung des PC-OS zu einer höheren Version leicht durchgeführt werden und es kann ein System für Koexistenz von Prozessen zur Unterdrückung von Hardwarekosten zur Verfügung gestellt werden.

Gemäß der Erfindung werden Daten zwischen Echtzeitprozessen oder Echtzeit- und Anwendungsprozessen ausgetauscht, so daß Echtzeit- und Anwendungsprozesse in Zusammenarbeit miteinander laufen können.

Gemäß der Erfindung wurde ein Echtzeitprozeß als ein Anwendungsprozeß entwickelt und zur Ausführung als ein Echtzeitprozeß dann geladen. Daher kann die Funktion, die zur Verfügung gestellt wird, leicht als ein Echtzeitprozeß programmiert werden.

Gemäß der Erfindung wird auch für Echtzeitprozesse ein Dienst zur Verfügung gestellt, der von dem Betriebssystem im wesentlichen für den Gerätetreiber von Eingabe/Ausgabe-Geräten zur Verfügung gestellt wird. Daher können Echtzeitprozesse, die als Anwendungsprozesse entwickelt wurden, sicher betrieben werden.

Gemäß der Erfindung wird die Zeit, in der Echtzeitprozesse die CPU-Kapazität beanspruchen, überwacht, so daß die Echtzeitprozesse die CPU-Kapazitäten nicht über eine gegebene Zeit hinaus besetzen. Daher können Echtzeitprozesse ausgeführt werden, ohne die Antwortfähigkeit auf den Nutzer des Anwendungsprozesses zu beeinträchtigen.

Gemäß der Erfindung wird ein Zugangskonkurrenzmechanismus unter dem Echtzeitsteuermechanismus zur Verfügung gestellt, so daß Echtzeitprozesse mit unterschiedlichen Eigenschaften zur gleichen Zeit unter einem Betriebssystem betrieben werden können.

Die vorstehende Beschreibung einer bevorzugten Ausführungsform der Erfindung wurde zum Zwecke der Erläuterung und Beschreibung dargestellt. Es ist nicht beabsichtigt vollständig zu sein oder die Erfindung durch die genaue offengelegte Form zu beschränken und Änderungen und Variationen sind im Lichte der obigen Lehre möglich oder können aus der Durchführung der Erfindung erworben werden.

Die Ausführung wurde gewählt und beschrieben, um

die Prinzipien der Erfindung zu erklären und ihre praktische Anwendung, um es dem Fachmann zu ermöglichen, die Erfindung in verschiedenen Ausführungsformen und mit verschiedenen Abänderungen zu nutzen, die für den besonderen beabsichtigten Gebrauch geeignet sind. In dieser Beschreibung ist auch Bezug genommen auf die prioritätsbegründende Voranmeldung JP 8-177 898 vom 8. Juli 1996 und die in ihr enthaltene Offenbarung ausdrücklich mit aufgenommen. Es ist beabsichtigt, daß der Rahmen der Erfindung bestimmt wird durch die hier angehängten Ansprüche und deren Äquivalente.

Patentansprüche

1. Echtzeitsteuersystem umfassend:
eine erste Steuervorrichtung zur Zuordnung von CPU-Kapazitäten zu Anwendungsprozessen und Steuerung der Anwendungsprozesse;
einen Treibereinbaumechanismus zum Einbau eines Gerätetreibers, um Eingabe/Ausgabe-Verarbeitungen für Geräte durchzuführen;
ein Echtzeitsteuermechanismus, der als ein Echtzeitsteuermechanismus auf Verarbeitung zugreift, die Echtzeitantwortfähigkeit auf einen Interrupt von einem Gerät erfordern, wie ein Gerätetreiber von einem Betriebssystem;
Echtzeitprozesse, die in einer Verarbeitungskorrespondenz mit den Geräten gebildet werden; und
eine zweite Steuervorrichtung, um CPU-Kapazitäten den genannten Echtzeitprozessen zuzuordnen und genannte Echtzeitprozesse zu steuern.
2. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß der genannte Echtzeitsteuermechanismus weiterhin eine Vorrichtung zur Kommunikation zwischen Echtzeitprozessen zur Übertragung von Daten zwischen den genannten Echtzeitprozessen aufweist.
3. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß der genannte Echtzeitsteuermechanismus weiterhin eine Vorrichtung zur Kommunikation zwischen Prozessen zur Übertragung von Daten zwischen dem Anwendungsprozeß, der unter dem Betriebssystem läuft, und dem genannten Echtzeitprozeß aufweist.
4. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß der genannte Echtzeitsteuermechanismus weiterhin einen Lademechanismus für Echtzeitprozesse zur Eintragung des unter dem Betriebssystem laufenden Anwendungsprozesses als genannten Echtzeitprozeß aufweist.
5. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß der genannte Echtzeitsteuermechanismus weiterhin eine Betriebssystemschutzvorrichtung aufweist, um eine Dienstfunktion zu ermöglichen, die durch das Betriebssystem für den Gerätetreiber zur Verfügung gestellt wird, um von den genannten Echtzeitprozessen genutzt zu werden.
6. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß der genannte Echtzeitsteuermechanismus weiterhin eine Vorrichtung zur Überwachung von Nutzungsrechten aufweist, um die Zeit, während der die zweite Steuervorrichtung die CPU-Kapazität von dem Betriebssystem beansprucht, zu messen und festzuhalten und um abhängig von dem Ergebnis der Messung die CPU-Kapazität zwangsweise an das Betriebssystem zurückzugeben.

7. Echtzeitsteuersystem gemäß Anspruch 1, dadurch gekennzeichnet, daß es weiterhin eine Vorrichtung zur Konkurrenzkontrolle aufweist, um ein Master-Slave-Verhältnis oder ein Prioritätsverhältnis unter den genannten Echtzeitsteuermechanismen einzustellen, wenn es eine Mehrzahl genannter Echtzeitsteuermechanismen gibt.

Hierzu 16 Seite(n) Zeichnungen

10

15

20

25

30

35

40

45

50

55

60

65

FIG. 1

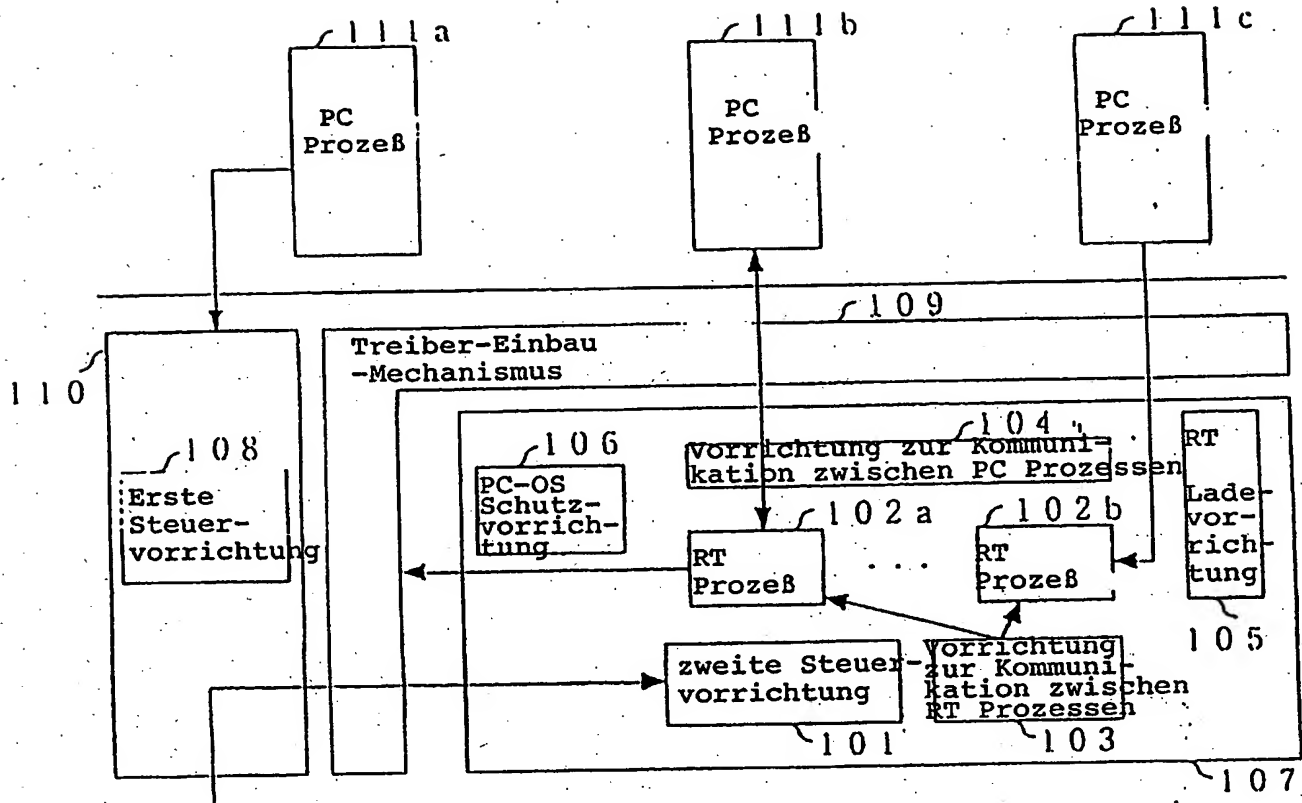


FIG. 2

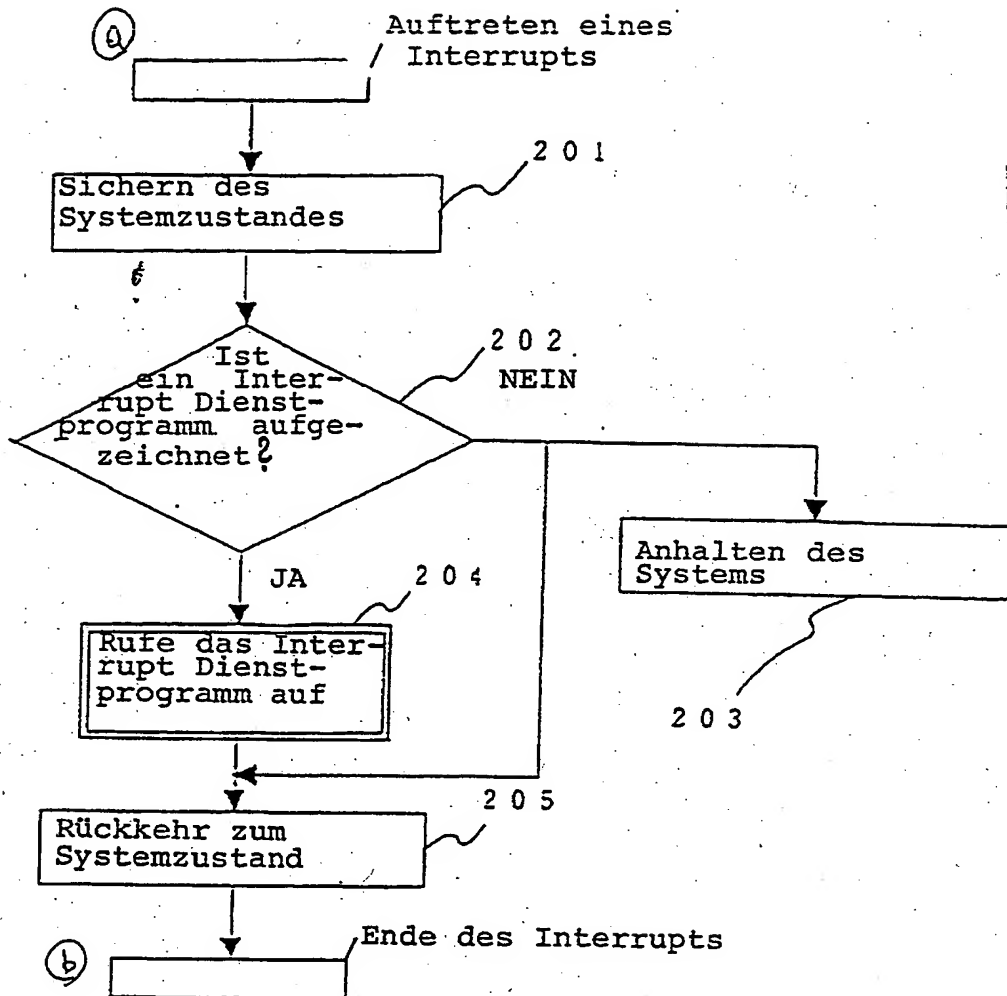
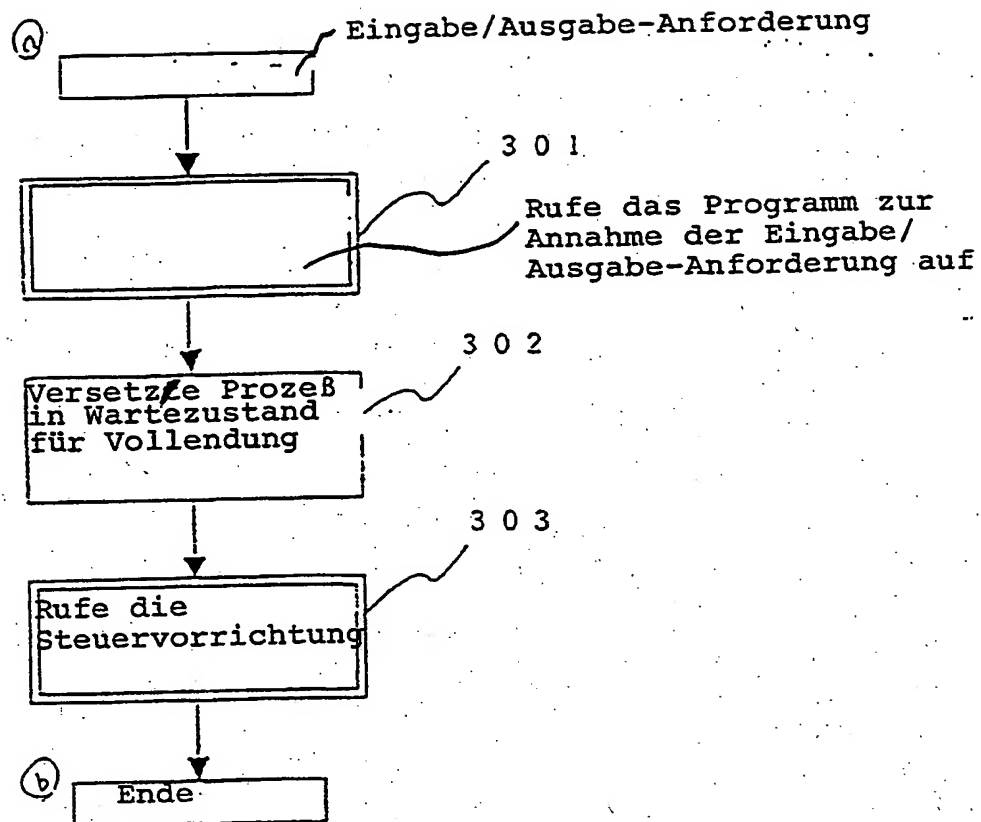


FIG. 3



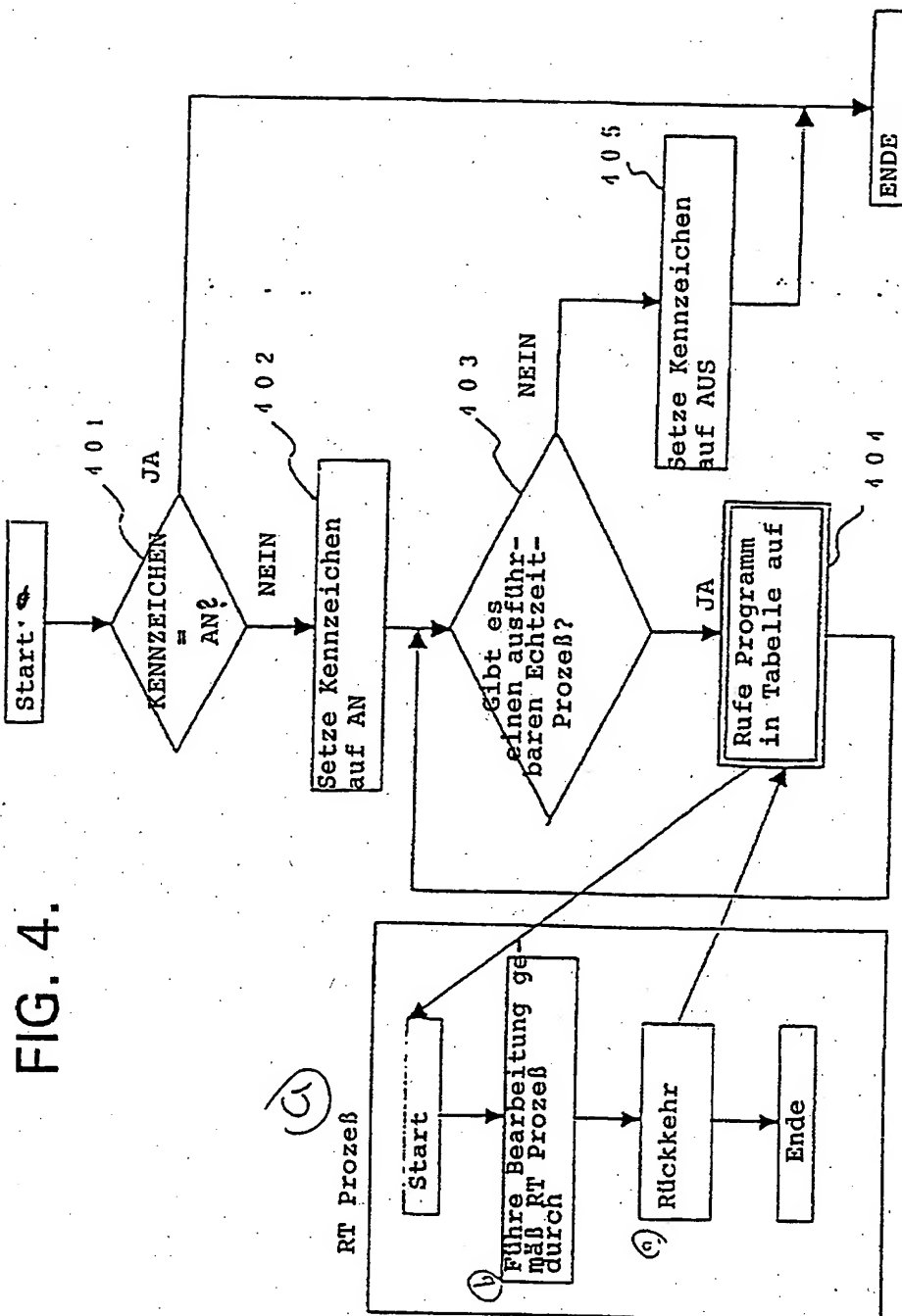


FIG. 5

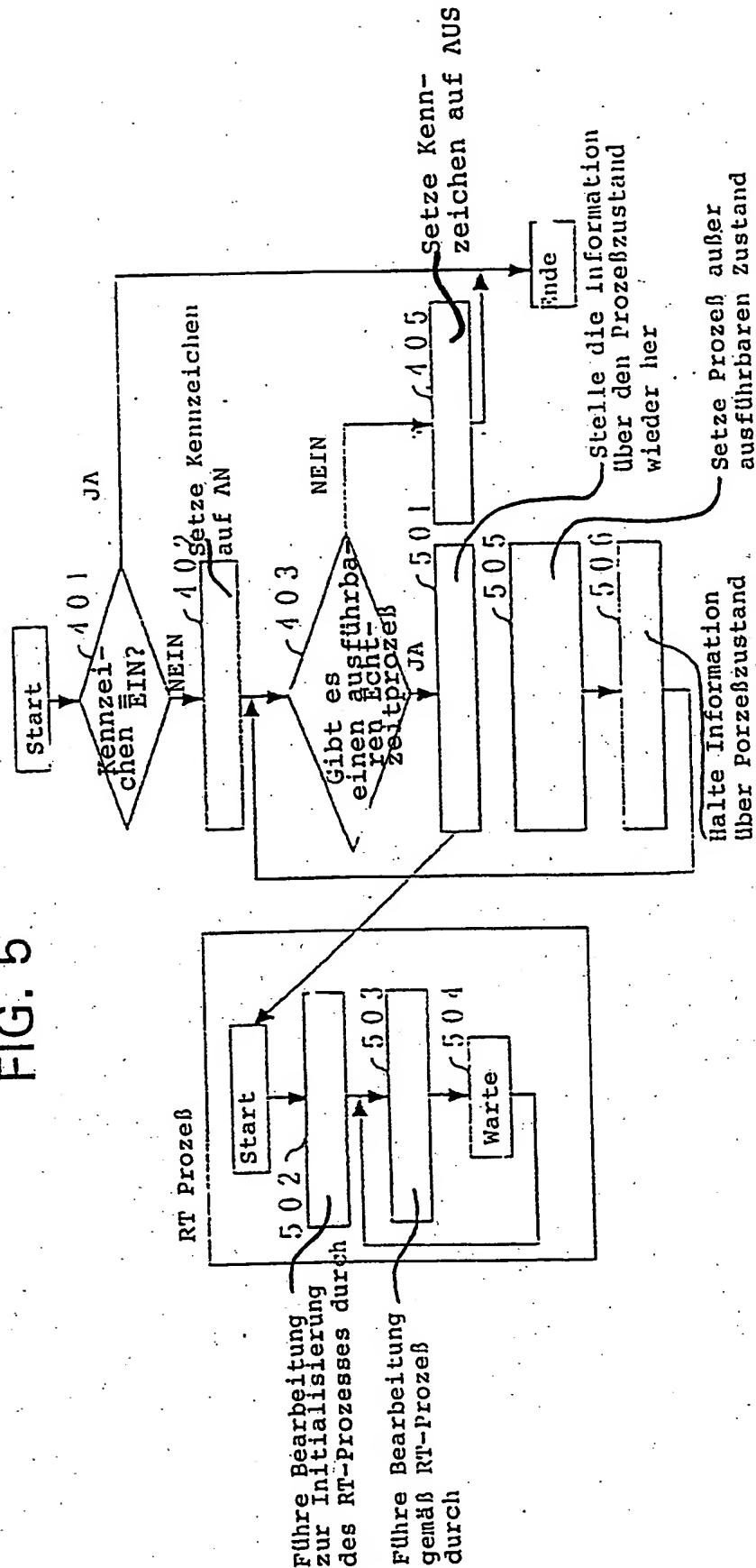


FIG. 6

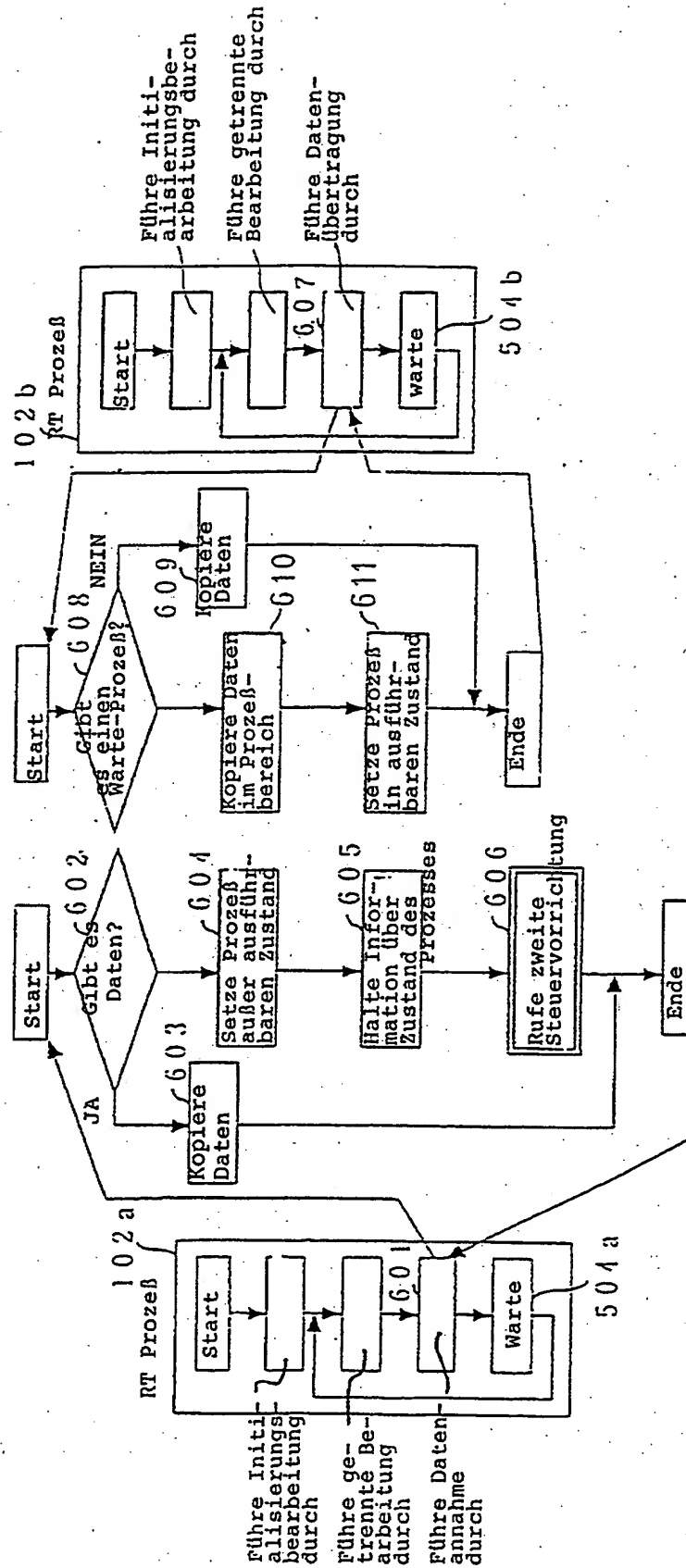


FIG. 7

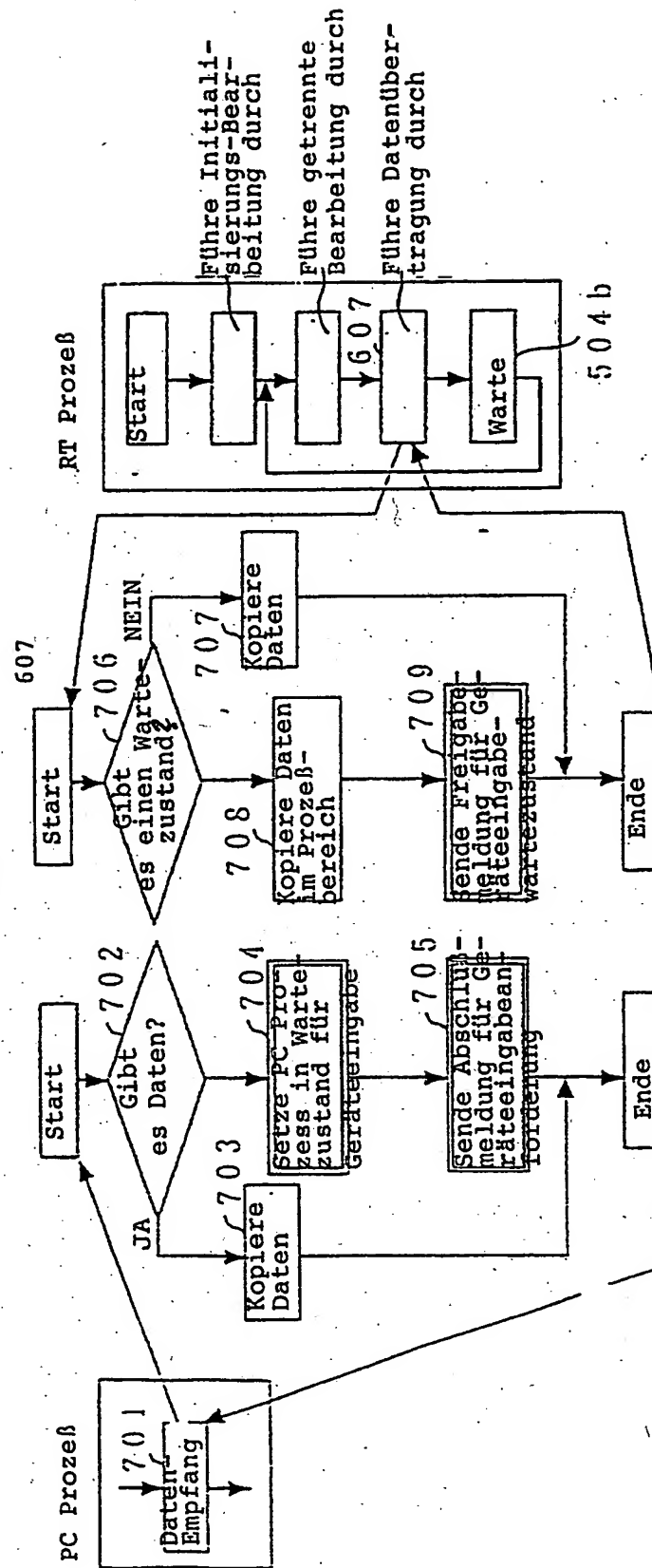


FIG. 8

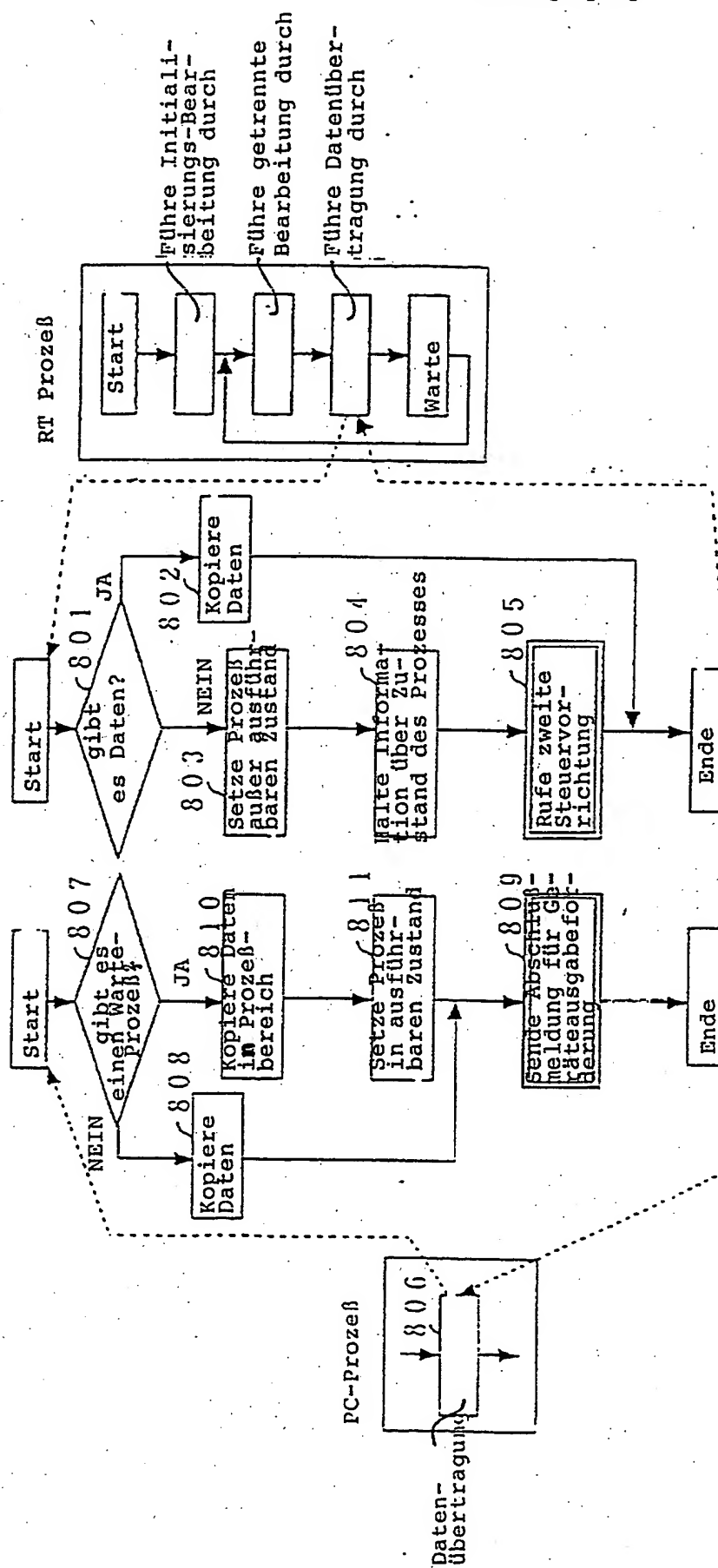


FIG. 9

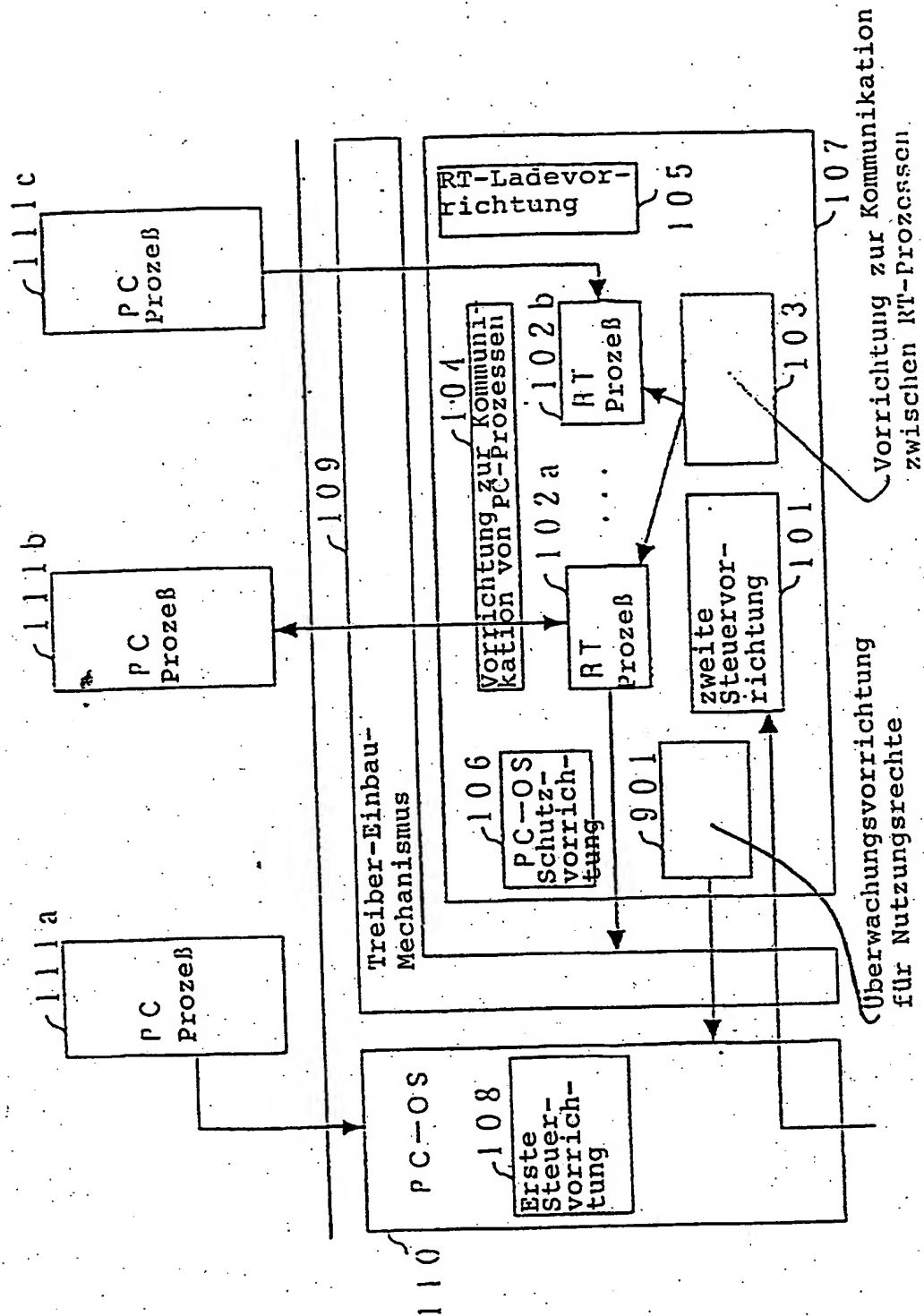


FIG. 10

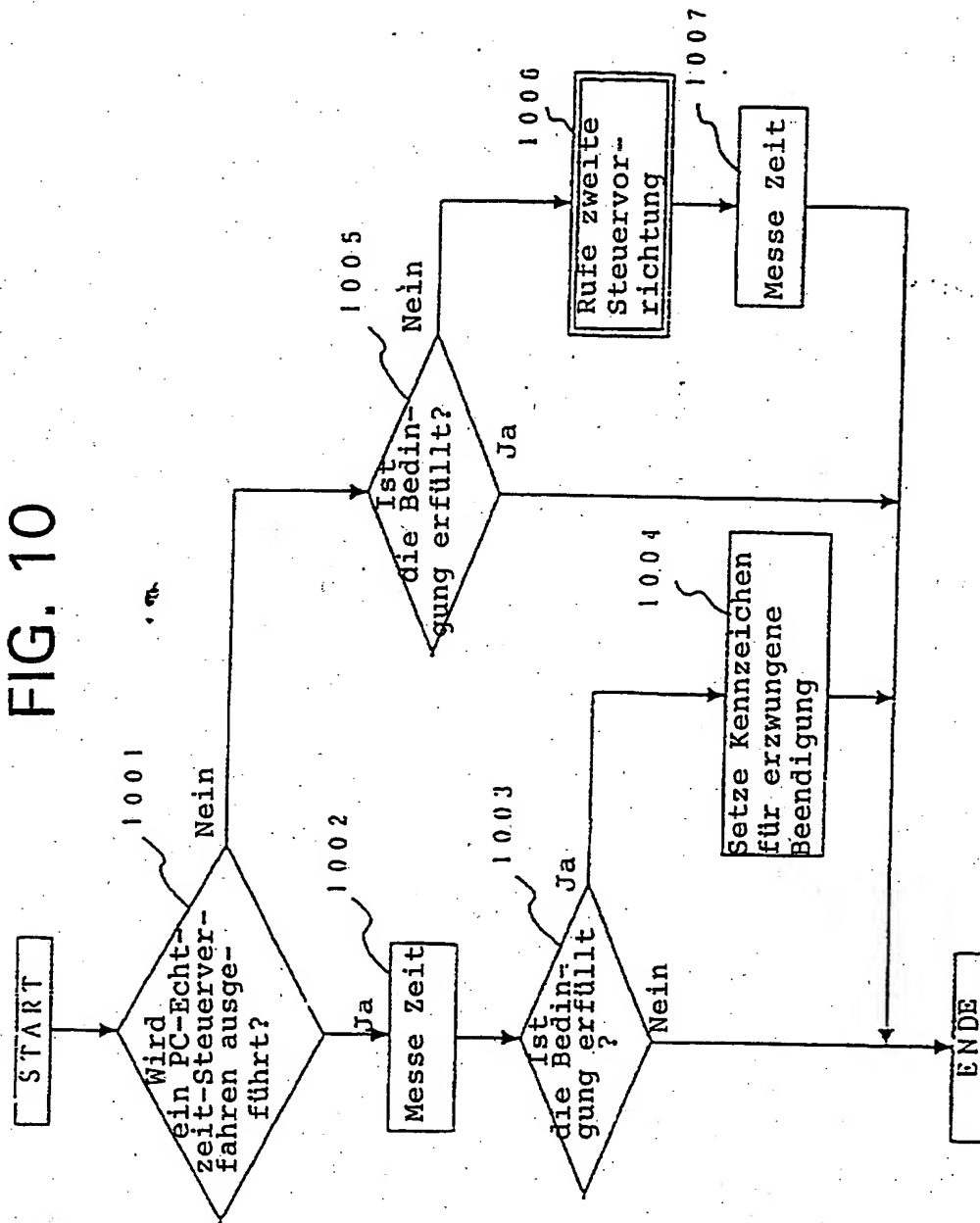


FIG. 11

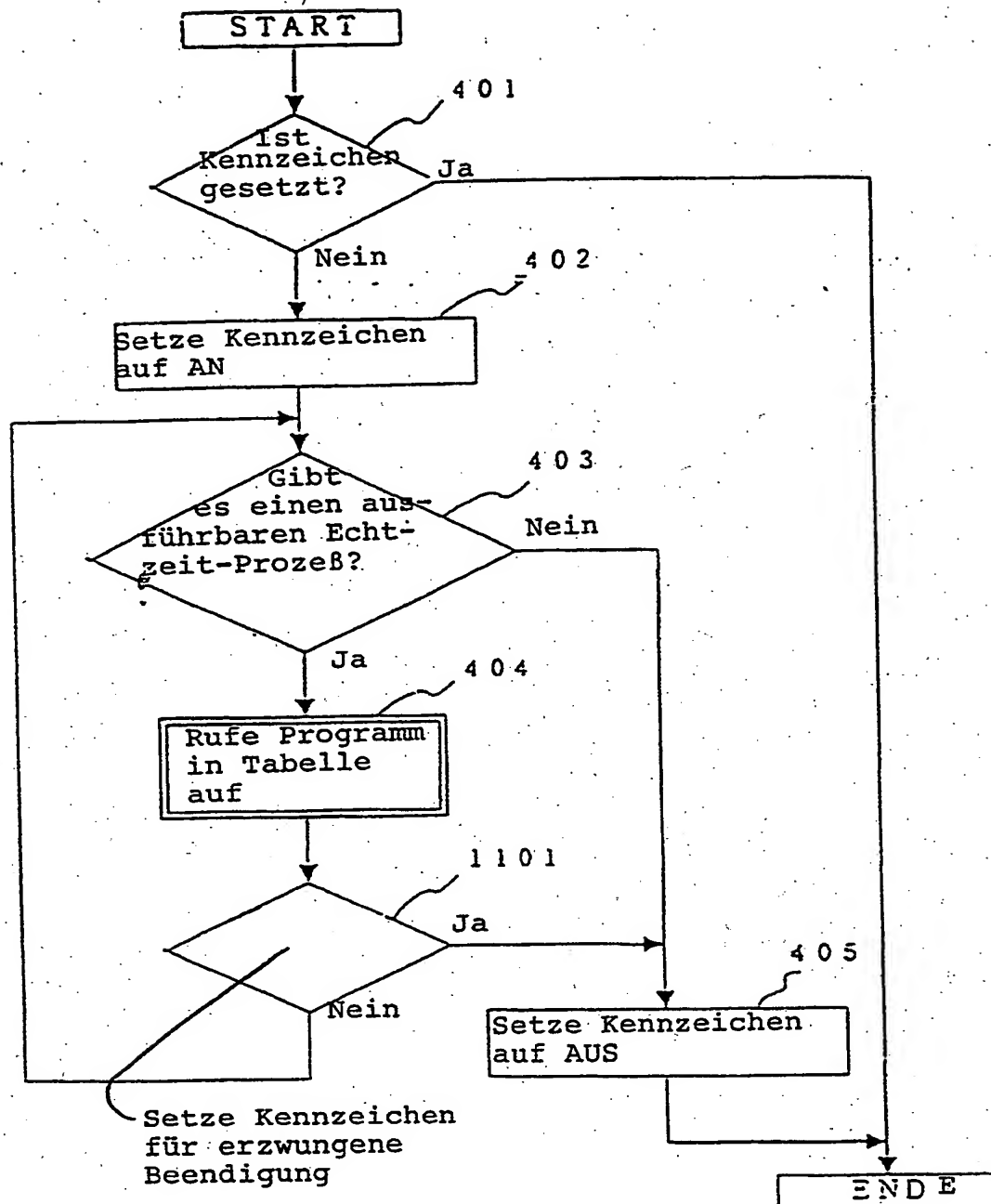


FIG. 12

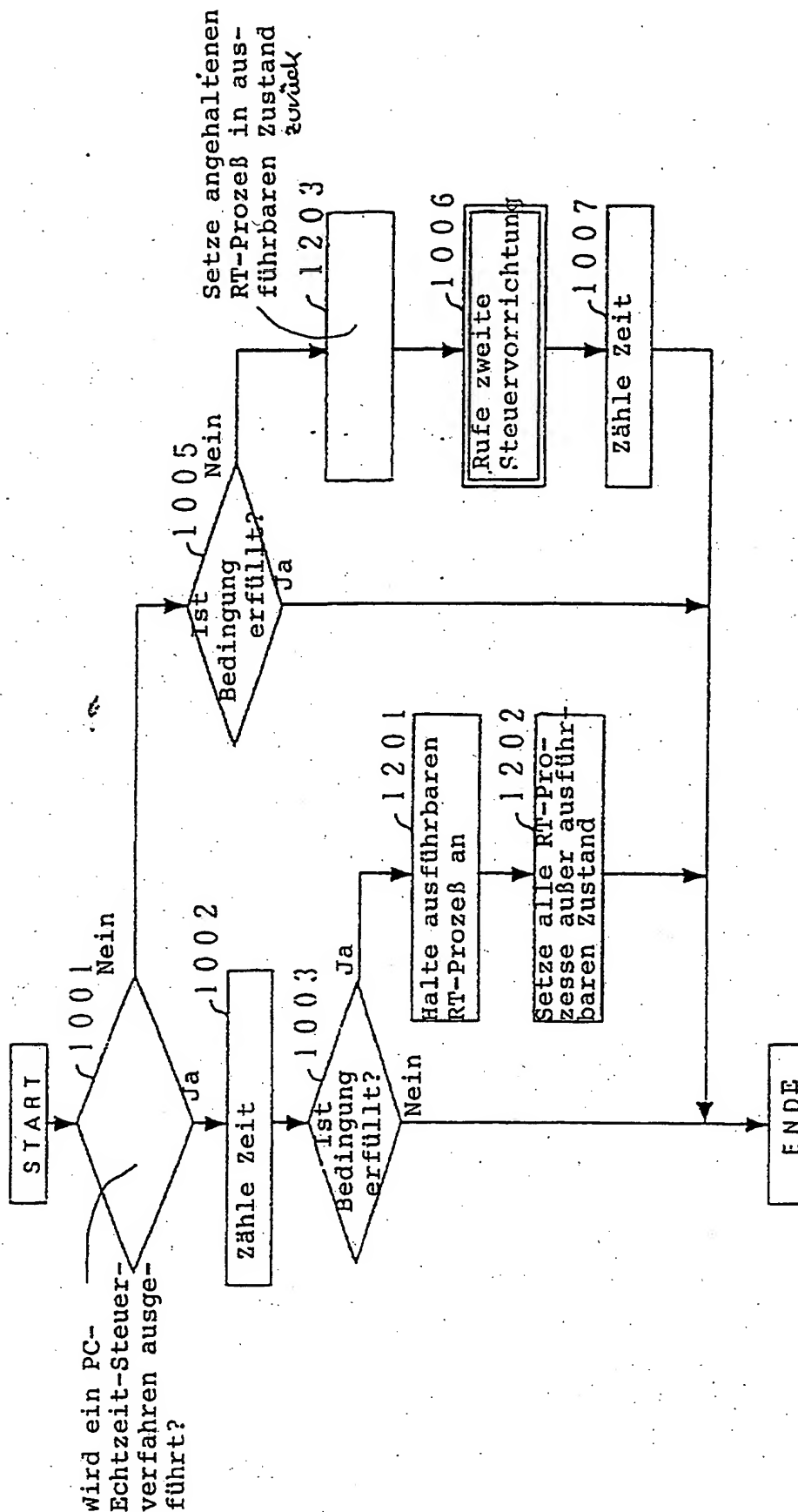


FIG. 13

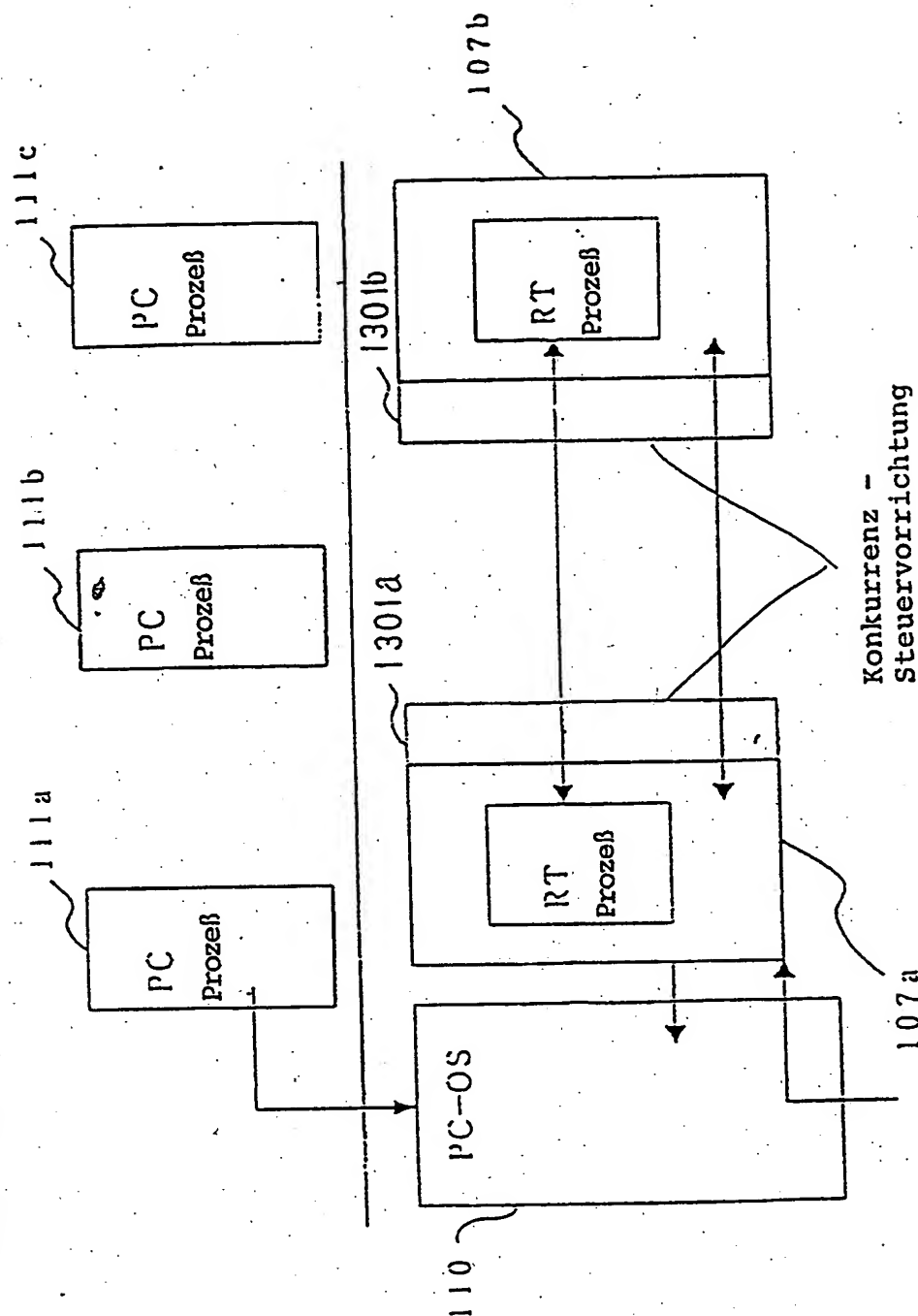


FIG. 14

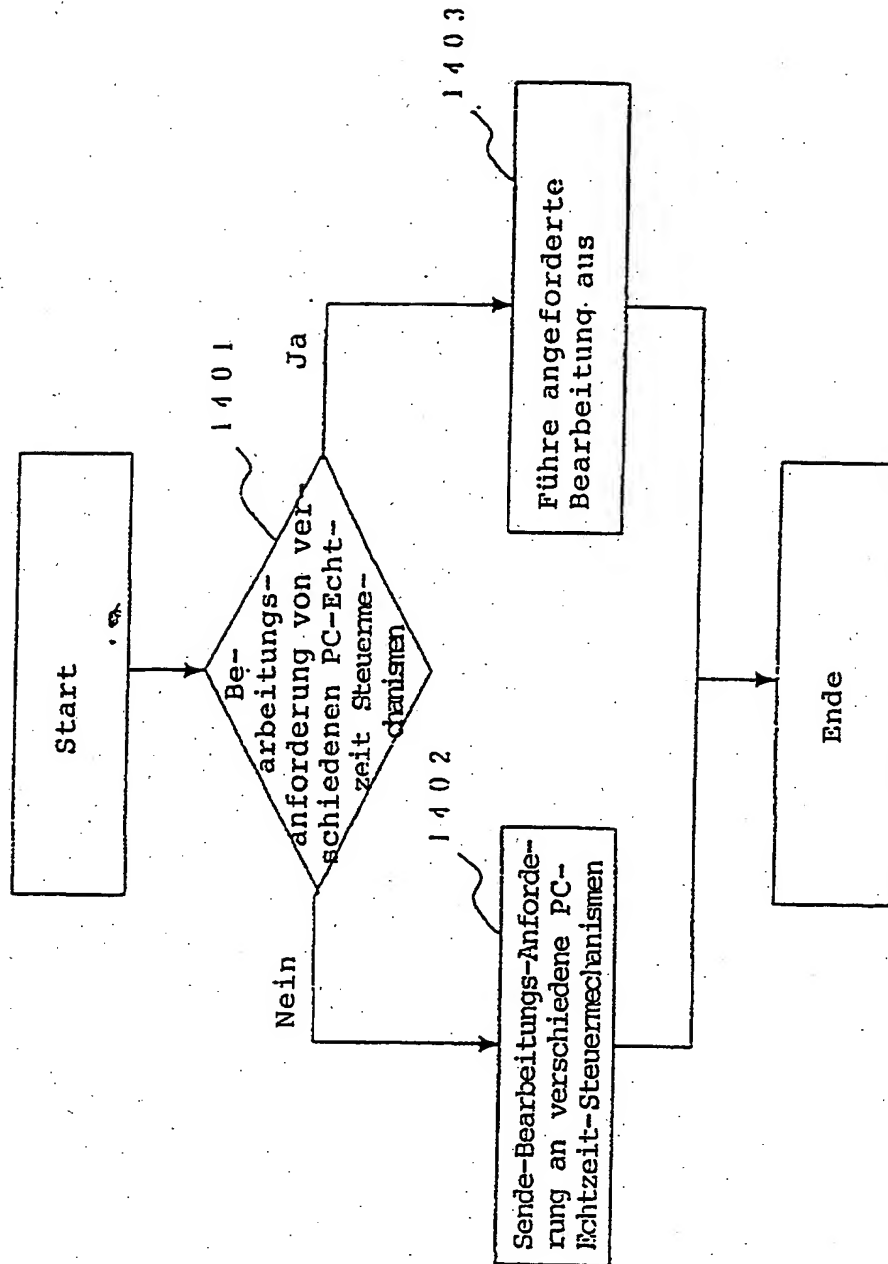


FIG. 15

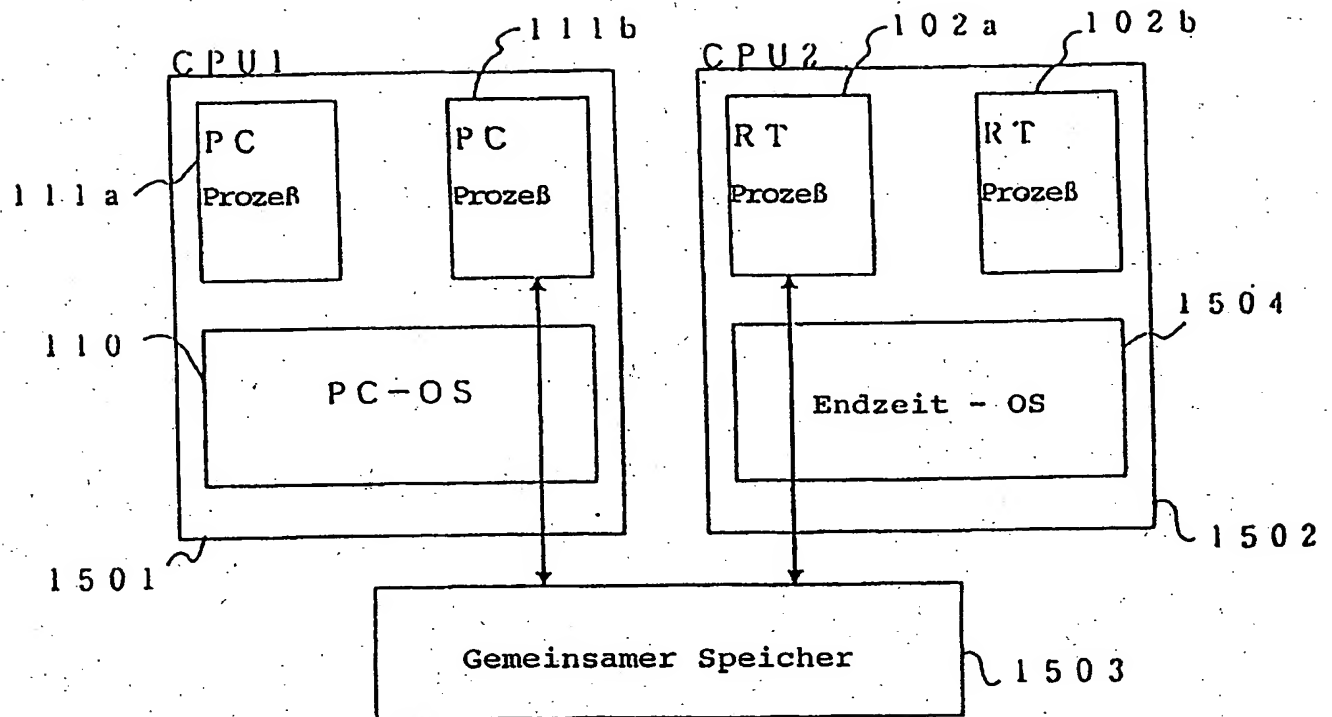
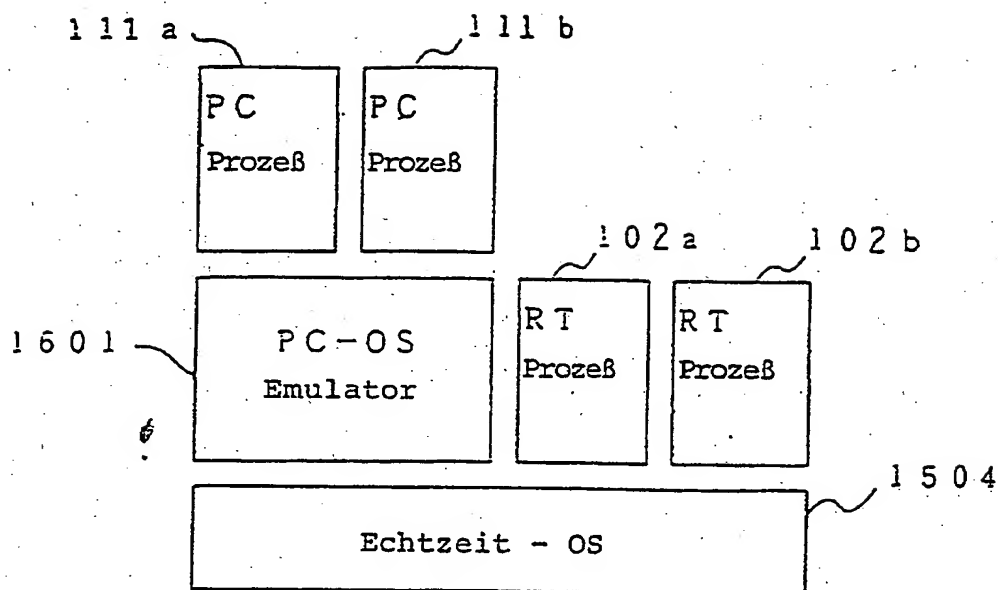


FIG. 16



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)